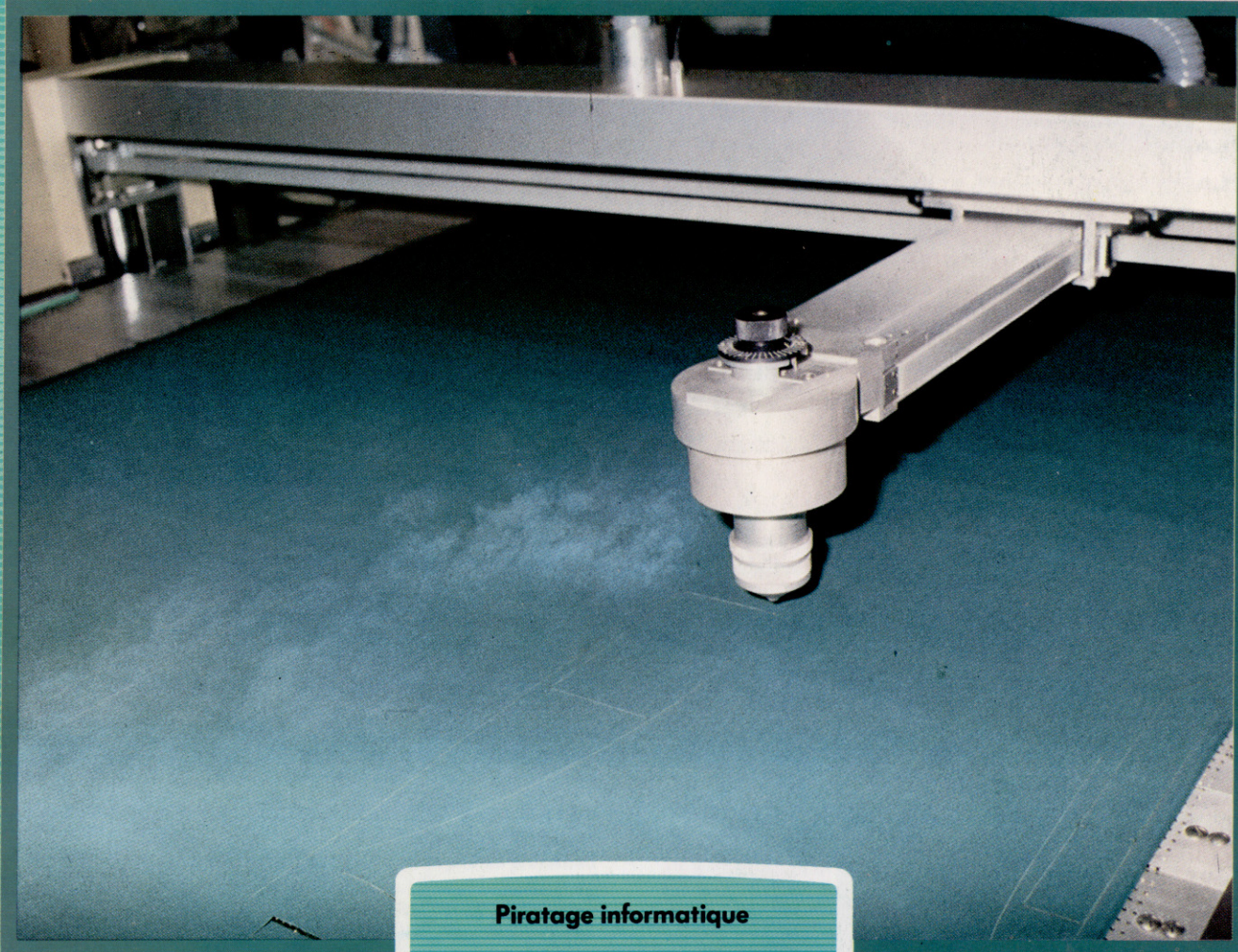


# abc

N° 48

COURS  
D'INFORMATIQUE  
PRATIQUE  
ET FAMILIALE

# INFORMATIQUE



Piratage informatique

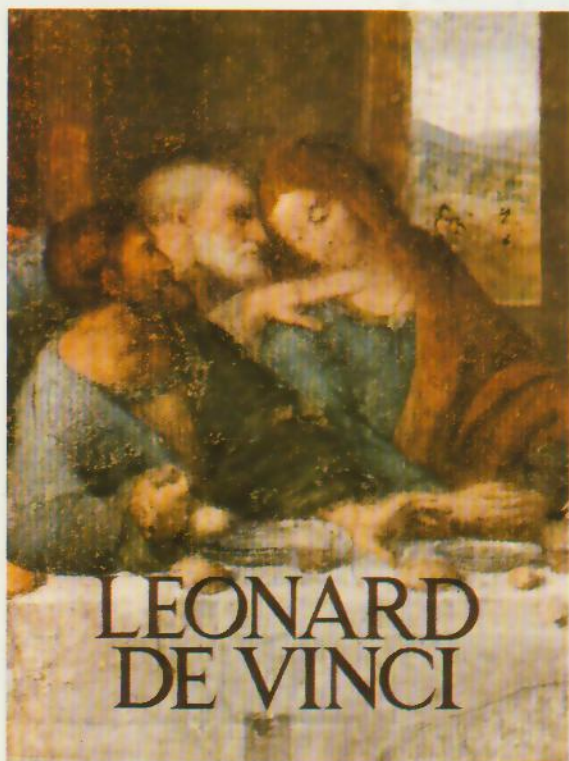
Micro et dactylo

Le Macintosh

La récursivité

EDITIONS  
**ATLAS**

Dans toutes les librairies

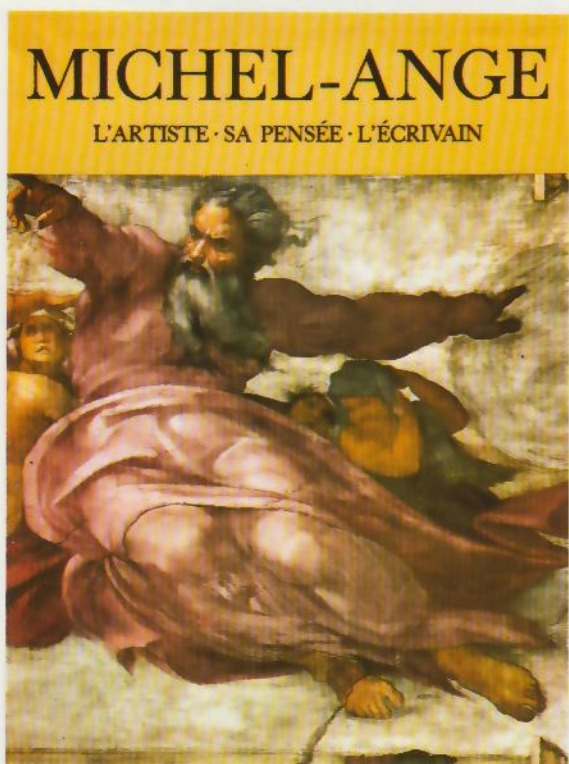


## Léonard de Vinci

Dessinateur, peintre, passionné de sciences naturelles, ingénieur, architecte, inventeur de machines de guerre, musicien, Léonard de Vinci fut vraiment un génie universel. Établi en tenant compte des dernières études et découvertes, cet ouvrage présente, en une suite de monographies largement illustrées, une vue complète de son œuvre.

*Un volume relié, présenté sous emboîtement luxueux. 542 pages illustrées. Nombreuses planches hors texte en couleurs. Format : 27 x 37 cm.*

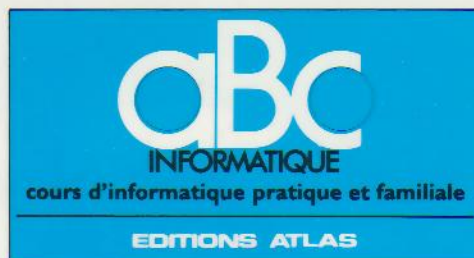
Dans toutes les librairies



## Michel-Ange

Toutes les connaissances et les attitudes de la critique artistique et de l'histoire de l'art sont présentées dans cet ouvrage, face à la personnalité exceptionnelle de Michel-Ange, artiste au génie multiple, à la fois sculpteur, peintre, architecte et poète, dont l'œuvre prodigieuse exprime de façon pathétique la lutte éternelle de l'esprit et de la matière.

*Un volume relié, présenté sous emboîtement luxueux. 612 pages illustrées. Nombreuses planches hors texte en couleurs. Format : 27 x 37 cm.*



Édité par ÉDITIONS ATLAS s.a., tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15. Tél. : (37) 28-10-10. Services administratifs et commerciaux : 3, rue de la Taye, 28110 Lucé. Tél. : (37) 28-10-10. Belgique : ÉDITIONS ATLEN s.a., Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, Montréal Nord.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, Mezzovico.

Realisé par EDENA s.a., boîte 185, 33, avenue du Maine, 75755 Paris Cedex 15.

Direction générale : J.-P. Turbergue. Service technique et artistique : F. Givone et J.-Cl. Bernar. Iconographie : J. Pierre. Correction : B. Noël.

Publicité : Anne Cayla. Tél. : 202-09-80.

### VENTE AU NUMÉRO

Les numéros parus peuvent être obtenus chez les marchands de journaux ou, à défaut, chez les éditeurs, au prix en vigueur au moment de la commande. Ils resteront en principe disponibles pendant six mois après la parution du dernier fascicule de la série (Pour toute commande par lettre, joindre à votre courrier le règlement, majoré de 10 % de frais de port.)

Pour la France, s'adresser aux services commerciaux des ÉDITIONS ATLAS, Z.I. de Lucé, 3, rue de la Taye, 28110 Lucé. Tél. : (37) 28-10-10.

Pour les autres pays, s'adresser aux éditeurs indiqués ci-dessous.

### SOUSCRIPTION

Les lecteurs désirant souscrire à l'ensemble de cet ouvrage peuvent s'adresser à :

France : DIFFUSION ATLAS, 3, rue de la Taye, 28110 Lucé. Tél. : (37) 35-40-23.

Belgique : ÉDITIONS ATLEN s.a., 55, avenue Huart-Hamoir, 1030 Bruxelles. Tél. : (02) 242-39-00. Banque Bruxelles-Lambert, compte n° 310-0018465-24 Bruxelles.

Canada : ÉDITIONS ATLAS CANADA Ltée, 11450, boulevard Albert-Hudon, Montréal Nord, H 1G 3J9.

Suisse : FINABUCH s.a., ÉDITIONS TRANSALPINES, zona industriale 6849 Mezzovico-Lugano. Tél. : (091) 95-27-44.

### RELIEZ VOS FASCICULES

Des reliures mobiles permettant de relier 12 fascicules sont en vente chez votre marchand de journaux.

**ATTENTION : ces reliures, présentées sans numérotation, sont valables indifféremment pour tous les volumes de votre collection. Vous les numéroterez vous-même à l'aide du décalque qui est fourni (avec les instructions nécessaires) dans chaque reliure.**

En vente tous les vendredis. Volume IV, n° 48.

ABC INFORMATIQUE est réalisé avec la collaboration de Trystan Mordrel (secrétariat de rédaction), Jean-Pierre Bourcier (coordination), Patrick Bazin, Jean-Paul Mourlon, Claire Rémy (traduction), Ghislaine Goullier (fabrication), Marie-Claire Jacquet (iconographie), Claire Bischoff (correction).  
Crédit photographique, couverture : Maillac/Réa.

Directeur de la publication : Paul Bernabeu. Imprimé en Italie par I.G.D.A., Officine Grafiche, Novara. Distribution en France : N.M.P.P. Tax. Dépôt légal : décembre 1984. 148412. Dépôt légal en Belgique : D/84/2783/27.

© Orbis Publishing Ltd., London.  
© Éditions Atlas, Paris, 1984.

### A NOS LECTEURS

En achetant chaque semaine votre fascicule chez le même marchand de journaux, vous serez certain d'être immédiatement servi, en nous facilitant la précision de la distribution. Nous vous en remercions d'avance.

Les Éditions Atlas



# Un œil noir te regarde

Les défenseurs de la loi ne sont pas tendres avec ceux qui parviennent à pirater les systèmes informatiques. Mais où en sont les techniques de surveillance informatique?



Steve Cross

Les techniques deviennent l'objet de champs d'application de plus en plus vastes, et sans doute y avez-vous déjà été soumis sans vous en rendre compte.

Certaines de ces techniques sont simplement établies à partir des fichiers relatifs aux automobiles, aux permis de conduire ou aux relevés de la Sécurité sociale. Mais le danger est que ces éléments soient rassemblés et confrontés avec d'autres, tout aussi innocents (compte en banque, loyer, abonnement à des revues, etc.). La simple juxtaposition de ces données peut permettre de tracer une sorte de portrait robot de n'importe quel citoyen. C'est l'une des objections majeures faites, il y a quelques années, au fameux projet SAFARI, mis au point par le gouvernement français.

La Commission nationale informatique et liberté (C.N.I.L.) a été spécifiquement créée pour lutter contre d'éventuels abus de ce type. Mais certains pensent que son efficacité sera

toujours réduite, en raison de l'extrême rapidité des progrès techniques en ce domaine.

Les systèmes de surveillance et de sécurité font fréquemment appel à ce que l'on appelle la « reconnaissance des formes » : l'ordinateur compare ce qu'il « voit » à des structures emmagasinées dans sa mémoire. Celle-ci doit donc être importante, et la manipulation des données réclame une très grosse puissance de traitement. Ces deux inconvénients sont désormais surmontés. Il est aujourd'hui possible, pour un prix raisonnable, de disposer des deux à la fois.

En Grande-Bretagne, par exemple, la société Logica a installé, dans les nouveaux bureaux de Scotland Yard à Londres, un système de traitement des empreintes digitales. Il a fallu quinze ans de travail pour réaliser le tout, qui peut stocker 650 000 empreintes différentes, plus 100 000 relevés partiels effectués au cours d'enquêtes criminelles. On les compare tout simplement aux données du fichier principal. Il faut pour cela des mini-ordinateurs, des systèmes de traitement de données très élaborés,

## Espionnage à domicile

Des dispositifs d'écoute peuvent être installés à peu près partout dans une pièce. Ce bureau d'un responsable commercial pourrait ainsi être « piraté ».

1. Le téléphone : le dispositif est logeable dans le récepteur, dans le corps de l'appareil ou, encore, dans la lampe qui est juste à côté.
2. La plante en pot : dans la terre, sous le pot, ou même en prenant l'apparence d'un insecte !
3. Le mur : l'appareil est coulé dans le plâtre, ou dissimulé derrière un revêtement de liège.
4. Le cadre accroché au mur : derrière lui, ou placé dans le système d'accrochage.
5. Le bureau : sous le bureau lui-même, ou dans un tiroir.



## Contre-espionnage

### Protection informatisée

De nombreux hommes d'affaires ou les personnes ayant à manipuler des informations confidentielles se protègent de l'espionnage électronique avec des dispositifs analogues à ce brouilleur téléphonique. Au lieu de parler dans le téléphone, l'usager tape au clavier son message que le brouilleur transmet ensuite par l'intermédiaire des lignes téléphoniques. Un système équivalent reçoit le message et le décode. Un synthétiseur de parole, sur appui d'un simple bouton, permet d'en donner une représentation audible.



### Message codé

De conception très semblable, ce procédé permet le transfert, sous forme codée, d'un message écrit à la main : l'opération se faisant en silence, les écoutes électroniques restent sans effet. Il est même possible de cette façon de transmettre des signatures.



### Analyseur de stress

Cet analyseur de voix permet de déterminer le niveau de stress qu'elle contient, et donne aussitôt son résultat sous forme numérique. C'est en fait un détecteur de mensonges très sophistiqué, et qui peut également indiquer si le sujet est nerveux ou tendu.



ainsi que des caméras et des moniteurs haute résolution. Malgré tout, il est encore impossible de manipuler plus de deux ou trois cents empreintes par jour.

Toujours en Grande-Bretagne, un système analogue est installé sur un pont qui enjambe

une autoroute. Des caméras, placées de façon à suivre tous les flots de circulation, enregistrent les numéros d'immatriculation des voitures qui passent. Les résultats obtenus de l'analyse des images peuvent ensuite être comparés à ceux obtenus à partir d'une liste de voitures recherchées (vols, etc.). S'il s'en trouve une dans le lot, un simple message radio permet aux patrouilles de police qui circulent sans arrêt sur l'autoroute de l'intercepter. Les autorités policières n'étaient guère désireuses d'évoquer publiquement un tel système; c'est un journaliste qui le premier eut vent de l'affaire.

Les progrès de la micro-informatique ont également permis la création de dispositifs de surveillance de plus en plus petits. Il est par exemple tout à fait possible de fabriquer des émetteurs radio de la taille d'un grain de riz et pourvus de contrôles électroniques très sophistiqués. C'est de cette façon qu'on met un téléphone sur écoute : l'appareil est alimenté par simple raccord sur les fils de courant et il ne se met en marche que lorsque quelqu'un parle dans le récepteur. D'autres modèles aussi minuscules ont leur propre source d'alimentation; placés dans une pièce, ils permettent d'entendre tout ce qui s'y dit : les conversations sont transmises à un récepteur situé à bonne distance. Là encore, ils ne fonctionnent qu'au son de la voix.

Mieux encore — nous sommes là en plein James Bond —, on peut aussi diriger un rayon laser vers une fenêtre. Les vitres de celle-ci vibrent d'une certaine façon chaque fois que l'on parle; une fois reflété, le rayon est donc brouillé par des interférences que l'on décompose et que l'on fait analyser par ordinateur.

Dans le domaine militaire, le problème est exactement inverse : les opérateurs ne doivent pas être entendus par d'autres. Mais la solution reste la même. Les émetteurs-récepteurs de campagne ont ainsi recours aux « sauts de fréquence » (passage contrôlé par ordinateur d'une fréquence à l'autre, selon un code prédéfini), afin d'échapper aux écoutes et au brouillage.

Pour le moment, seuls de très gros systèmes peuvent réellement assurer des tâches de surveillance. Mais les avancées technologiques récentes permettent de penser que l'analyse des empreintes digitales (et bientôt, peut-être, des visages) sera sous peu entièrement automatisée et, de surcroît, bon marché.

Déjà des cars de police sont équipés d'un ordinateur capable d'extraire des données de n'importe quel fichier (scolaire, criminel, médical, etc.). Il suffit pour cela d'insérer dans l'appareil une carte officielle plastifiée. La machine accepte les empreintes digitales et la photographie d'un suspect, qu'elle compare à un fichier central.

Vision paranoïaque de l'avenir? Peut-être. Mais de nombreux défenseurs de la loi et de l'ordre y sont plus que favorables. Et n'oublions pas le fameux slogan de Big Brother : « L'esclavage c'est la liberté/La liberté c'est l'esclavage. »



# Avoir la touche

**La maîtrise de la dactylographie est devenue essentielle, et pas seulement pour les employés de bureau. Les programmeurs, les journalistes doivent taper rapidement et sans faire d'erreur.**

La possibilité de taper des mots complets sur papier ou sur écran, plutôt que d'avoir à taper laborieusement lettre par lettre, accélère grandement l'exécution d'un travail et réduit de beaucoup les risques d'erreurs.

Il existe diverses méthodes d'enseignement de la dactylographie (des programmes d'ordinateur lui sont mêmes consacrés). Quelle que soit la méthode d'enseignement, quatre éléments sont essentiels pour devenir un dactylographe efficace. Les voici :

- La maîtrise du clavier.
- L'habitude de garder les yeux sur l'écran, ou sur le papier.
- La rapidité.
- La précision.

Ces capacités sont développées lentement, au moyen d'exercices, jusqu'à ce que le dactylographe ait atteint un certain niveau d'efficacité.

La répétition est encore la meilleure méthode d'apprentissage; il faut enchaîner les mêmes lettres jusqu'à ce que la frappe devienne automatique.

La disposition standard du clavier de langue française est nommée AZERTY, parce que ces lettres apparaissent sur la deuxième rangée de touches du clavier. Les symboles et les lettres du clavier sont placés en fonction de leur fréquence d'utilisation, et chaque rangée du clavier est divisée en deux parties, l'une pour la main droite, et l'autre pour la main gauche.

L'apprentissage de la frappe rapide débute généralement par la maîtrise des huit touches de la rangée du milieu. Ces touches sont dites « touches repères » parce que les doigts reviennent sur ces touches après avoir frappé d'autres touches sur le clavier. Les touches repères du côté gauche du clavier, et donc destinées à la main gauche, sont q, s, d, f. Les touches repères

#### Une méthode efficace

Des cours de dactylographie pour débutants s'avèrent parfois très efficaces. Cette photo illustre comment étaient exploitées les techniques audiovisuelles il y a peu de temps. (Cl. Doriant/Groupe Pigier.)





du côté droit du clavier, pour la main droite, sont j, k, l, m. Dès que l'usage de ces touches a été maîtrisé, le dactylographe apprend l'emplacement des autres touches en fonction de leur position par rapport aux touches repères, et les atteint par extension des doigts. Le débutant est toujours surpris de la fréquence d'utilisation du petit doigt. On se sert de tous les doigts pour obtenir une frappe rapide, et le petit doigt doit « couvrir » un certain nombre de touches, comme tous les autres. Dès que ces touches ont été apprises, le dactylographe passe à la touche espace (pressée avec les pouces), et à la touche majuscule.

### Maître micro...

L'apprentissage de la dactylographie passe aujourd'hui par un des outils les plus modernes : le terminal informatique. (Cl. Doriant/Groupe Pigier.)

Malheureusement pour l'utilisateur d'ordinateur domestique, les touches numériques sont rarement abordées dans les cours ou dans les manuels de dactylographie. Pour le programmeur, les touches numériques représentent une partie spécialement importante du clavier. La frappe rapide des touches numériques est assez

facile à apprendre lorsque la notion de touches repères a été parfaitement maîtrisée. Les doigts de la main gauche effectuent simplement une extension au-dessus des touches alphabétiques pour frapper les touches 1 à 6, tandis que les doigts de la main droite se chargent des autres chiffres et des symboles qui suivent.

Il est possible d'étendre l'approche des touches repères d'un ordinateur domestique possédant des caractères graphiques qui peuvent être sollicités directement au clavier. On peut envisager, en apprenant l'emplacement des symboles graphiques, de les incorporer dans un programme de frappe rapide.

L'habitude de ne regarder que l'écran (ou le papier) s'acquiert durant les exercices faits sur le clavier. Cela représente l'aspect le plus important de la frappe rapide, et souligne nettement les lacunes de la frappe à deux doigts. Couvrir les touches avec du ruban adhésif ou des capuchons spéciaux aide beaucoup le débutant, car il





n'est plus tenté de regarder ses mains. A cet égard, l'ensemble clavier-moniteur est préférable à une machine à écrire et à une feuille de papier. L'écran se trouve au niveau des yeux et la tentation de jeter un coup d'œil vers le bas est moins forte. Les erreurs peuvent aussi être corrigées immédiatement, ce qui assure une plus grande précision.

La précision est une question de pratique et de concentration. Vous devez frapper la touche d'un geste rapide et ferme, dirigé au centre de la touche; les touches doivent être frappées de façon régulière, ce qui crée un rythme qui, avec le temps, devient naturel. Éventuellement, la frappe de la mauvaise touche ou l'utilisation du mauvais doigt est repérée « instinctivement », ce qui contribue grandement à acquérir de la vitesse.

La vitesse est mesurée au moyen de tests et d'exercices, et elle est apprise uniquement après les trois premières étapes dont nous venons de parler. Les meilleurs dactylographes peuvent atteindre des vitesses de 100 mots à la minute (m.p.m.). Pour le débutant, il est raisonnable de viser la vitesse de 30 m.p.m.

Il existe plusieurs manuels d'enseignement différents, et de nombreuses méthodes ont été conçues. Elles font toujours appel aux principes que nous avons énoncés plus haut. De grandes écoles de secrétariat distribuent encore des manuels qui ont été écrits il y a plus de trente ans. Puisque des mots courts et fréquemment employés sont utilisés dans ces exercices initiaux, l'orthographe ne présente aucune difficulté; cela permet au débutant de se concentrer sur l'acquisition de la technique.

Des cours permettent aussi d'apprendre la dactylographie à des enfants de onze ans en moins de 10 heures. Avec cette méthode, le débutant détache la page du manuel et tape la réponse sous l'exercice. Cela correspond à la méthode traditionnelle, dans laquelle l'étudiant copie exactement une série d'exercices. Ceux-ci sont accompagnés de dessins de mains montrant la position correcte sur le clavier.

Certaines écoles ont parfaitement intégré les méthodes modernes et les méthodes traditionnelles de l'enseignement de la dactylographie. Elles impliquent l'utilisation de voyants clignotant et de cassettes enregistrées. Ce système d'enseignement audiovisuel simule la simultanéité de la vision, de l'audition et de la réaction. Sur un grand panneau situé au-dessus de la tête de l'élève, une lettre clignote. La bande est synchronisée avec le panneau, et la voix pré-enregistrée de l'instructeur dit « maintenant », ce qui définit la vitesse de frappe. Tout en progressant, l'instructeur sollicite la frappe de la lettre de plus en plus rapidement. On prétend que cette technique relève un peu du lavage de cerveau. Des professeurs reconnaissent qu'ils ne comprennent pas vraiment pourquoi cette méthode fonctionne. Cependant, le nombre de clients satisfaits qui ont appris à dactylographier efficacement et sans effort indique que le système fonctionne, quelle qu'en soit la raison.

## Aperçus des logiciels

Les programmes d'enseignement de dactylographie sur ordinateur sont basés sur des jeux ou sur un texte. Les programmes basés sur des textes utilisent surtout des exercices répétitifs. La plupart des leçons de ce type de programme se présentent sous la forme d'exercices écrits que l'utilisateur doit copier exactement. Les programmes basés sur des jeux utilisent des graphiques, des actions rapides et des sons.



Ian McKinnell

### Type Invaders

Éditeur : Carswell Computers.  
Machine : BBC, modèle B.

C'est un programme de jeu destiné aux personnes ayant des notions de base en dactylographie. Son affichage graphique simple présente des lettres qui doivent être détruites en les tapant correctement. Les mots et les lettres qui ont été ratés attaquent de nouveau et peuvent éventuellement briser des lignes de défense et occuper votre terrain. Il y a dix niveaux de jeu différents, allant des majuscules à des mots de cinq lettres comportant des majuscules, aux minuscules et aux symboles. Il y a aussi trois vitesses : facile, rapide et très rapide. Même l'option « très rapide » semblera facile aux dactylographes expérimentés. Cependant, ce programme est agréable et pratique pour les dactylographes lents ou moyens.

### Sprintyper

Éditeur : Micro Software International.  
Machine : Commodore Vic-20.

Ce programme texte vise à accroître la vitesse et la précision des débutants et des dactylographes avancés. Il offre une bibliothèque de 356 635 phrases d'exercices. Au début, une phrase facile est affichée à l'écran, celle-ci doit être tapée le plus rapidement possible. Chaque erreur est signalée par un son grave qui ne s'interrompt qu'après correction de cette erreur. Dès que la phrase a été tapée correctement, le temps de frappe, le nombre d'erreurs et un tableau de performances apparaissent à l'écran. Sprintyper est essentiellement un test de vitesse qui présente peu d'intérêt pour le débutant.

### Typing Tutor II

Éditeur : Microsoft.  
Machines : Apple IIe et Apple IIe+.

Pour exécuter ce programme, vous avez besoin d'Applesoft en ROM, de 48 K de mémoire, d'un lecteur de disquettes et du DOS 3.3. Il s'agit d'un programme texte piloté par menus, comportant une combinaison de leçons, de paragraphes d'exercice et de tests de vitesse. La fonction la plus importante de Typing Tutor est le système dit « Contrôle de rapidité de réponse », qui vérifie la frappe 100 fois par seconde, détectant la moindre pause pouvant survenir. Les débutants commencent par taper des lettres. Après s'être familiarisés avec ces lettres, et lorsque la vitesse de frappe équivaut à 30 mots à la minute, de nouvelles lettres sont présentées à une vitesse plus grande. Pour les dactylographes expérimentés, le rapport de progression, relatif au paragraphe, donne le nombre d'erreurs commises, les touches sur lesquelles ces erreurs ont été faites, la vitesse et la précision. Nous recommandons fortement ce programme à tout niveau. Il est au départ un peu difficile à utiliser, et nous conseillons aux utilisateurs de lire attentivement la documentation avant de commencer à se servir de ce programme.

# Plus éblouissant

## Pour notre projet de programmation destiné au BBC Micro et à l'Electron, nous avons écrit les routines principales de notre « Jeu de Mines ». Ajoutons des raffinements.

Le premier élément nouveau sera d'écrire une routine « tireur embusqué » qui mettra en œuvre à l'écran un tireur faisant feu sur le détecteur de mines et sur le personnage. Un tir sera représenté par une ligne haute résolution traversant l'écran à partir de la limite gauche du champ de mines vers la droite. Afin d'introduire un élément aléatoire dans le tir, nous choisirons les coordonnées de départ et d'impact selon la fonction RND. Les valeurs pour *x* début et *x* fin sont assignées dans la procédure d'initialisation des variables. La différence entre ces deux valeurs est de 1 024 unités. La ligne de tir doit se composer de segments courts qui testeront, après être tracés, la zone encore à parcourir dans le but de trouver la couleur logique 1 (avec la commande POINT). La séquence de segments se poursuivra ainsi jusqu'à l'autre bout de l'écran si le tir n'a rien rencontré.

Il nous faut maintenant décider de la longueur d'un segment. Plus elle sera petite, plus le temps sera long pour tracer la ligne. Par contre, si le segment est trop long, il se peut que la détection n'ait pas lieu. Une position ayant l'équivalent de 64 unités graphiques de largeur, un segment de 32 unités graphiques (c'est-à-dire la moitié d'une position) semble indiqué. Si nous choisissons un segment de 32 unités dans la direction des *x* (*dx*), nous pouvons tracer la ligne sur un total de  $1\ 024 : 32 = 32$  segments. Si les ordonnées de départ et d'arrivée sont déterminées au hasard, la longueur d'un segment dans la direction des *y* (*dy*) sera obtenue en divisant la différence entre les deux valeurs par 32.

Notre dernier problème est d'effacer la ligne une fois qu'elle est tracée. Nous trouvons la solution dans le concept des couleurs logiques du BASIC BBC et sa propriété d'effectuer des opérations logiques entre elles. Dans le mode 5, il y a quatre couleurs logiques qui, sauf modifications, sont :

Couleur logique	0	1	2	3
Équivalent binaire	00	01	10	11
Couleur obtenue	noir	rouge	jaune	blanc

En utilisant GCOL, nous pouvons effectuer diverses opérations logiques entre les couleurs

tracées et les couleurs qui existent au moment du tracé. La commande a deux paramètres; le deuxième donnant la couleur logique à tracer. Le premier nombre donne la méthode du tracé :

GCOL0	Trace dans la couleur spécifiée
GCOL1	Effectue une opération OU
GCOL2	Effectue une opération ET
GCOL3	Effectue une opération OU EXCLUSIF
GCOL4	Effectue une opération NON sur la couleur présente

Quelques exemples vous montreront le fonctionnement de la commande. Si la couleur blanche (couleur logique 3) est présente sur la position à tracer et que nous la voulons en rouge (couleur logique 1), les divers modes d'utilisation de GCOL donneront les résultats suivants :

GCOL0,1	Efface le blanc et trace en rouge	
GCOL1,1	OU entre rouge et blanc pour tracer en blanc	rouge 01 blanc OU 11 blanc 11
GCOL2,1	ET entre rouge et blanc pour tracer en rouge	rouge 01 blanc ET 11 rouge 01
GCOL3,1	OU EXCLUSIF entre rouge et blanc pour tracer en jaune	rouge 01 blanc XO 11 jaune 10
GCOL4,1	NON sur le blanc pour tracer en noir	blanc 11 noir 00

En quoi cela nous aide-t-il à effacer les lignes de tir? Nous pourrions aussi tracer la ligne en blanc et repasser dessus en noir pour l'effacer. Mais s'il y avait déjà quelque chose sous la ligne, comme une mine, cela laisserait un trou. Nous pouvons cependant utiliser un OU exclusif (XOU) entre le rouge et la couleur existante pour chaque point croisé. Lorsque la zone est blanche, nous obtenons un segment de couleur jaune. Si nous traçons à nouveau la même ligne par un opérateur OU exclusif avec le rouge, le résultat final sera :

rouge	01
jaune	10
XOU	—
blanc	11

Ainsi, la couleur d'origine est restituée. Vous pouvez vérifier que deux tracés successifs de la même ligne par deux XOU reviennent toujours à rendre la couleur d'origine. Vous pouvez utiliser cette propriété pour effacer les lignes de tir.





Si vous tracez une ligne par XOU, et si vous retracez la même ligne encore par XOU, vous effacerez la ligne et restituerez les couleurs d'origine. Voici le listage complet de la procédure de tireur embusqué :

```
3110DEF PROCtireur
3120ydebut=RND(750)+220
3130yfin=RND(750)+220
3140dx=32:dy=(yfin-ydebut)/32
3150GCOL 3,3
3160PROCligne
3170IF POINT(x,y)=1 THEN PROCexplosion(x,y)
    ELSE PROCligne
3180 ENDPROC
```

Et voici le listage de la procédure de ligne :

```
3450DEF PROCligne
3460SOUND0,-8,4,5
3470x=xdébut:y=ydébut
3480MOVE x,y
3490REPEAT
3500DRAW x,y
3510x=x+dx:y=y+dy
3520UNTIL x>xfin OR POINT(x,y)=1
3530ENDPROC
```

Il est possible de générer des sons relativement complexes avec le BBC Micro. Pour ceux d'entre vous attirés par les effets sonores, nous allons ajouter une note musicale au programme. Pour que les choses soient simples, nous n'utiliserons qu'un seul canal. La mélodie peut être jouée en spécifiant la fréquence et la durée de chaque note :

```
4090DEF PROCmusique
4100REM ** 1er ACCORD **
4110SOUND1,-8,213,5
4120SOUND1,-8,209,5
4130SOUND1,-8,213,5
4140SOUND1,-8,209,5
4150SOUND1,-8,213,5
4160SOUND1,-8,193,5
4170SOUND1,-8,205,5
4180SOUND1,-8,197,5
4190REM ** 2 ième ACCORD **
4200SOUND1,-8,185,20
4210SOUND1,-8,165,5
4220SOUND1,-8,185,5
4230SOUND1,-8,193,20
4240REM ** 3 ième ACCORD **
4250SOUND1,-8,165,5
4260SOUND1,-8,193,5
4270SOUND1,-8,197,20
4280ENDPROC
```

**Page titre :** nous pouvons associer l'idée du tracé par XOU et celle du tracé relatif de points, pour produire un effet intéressant de séquence de titre. Cette procédure trace le mot MINES par l'intermédiaire du mode graphique haute résolution. Chaque nouvelle ligne tracée pour le mot l'est par rapport à la ligne précédente, de sorte que nous pouvons positionner le mot en entier n'importe où à l'écran, simplement en spécifiant le point de départ. Si nous traçons le mot et si nous le retraçons par XOU avant de nous déplacer vers le haut pour recommencer à l'écrire, nous créons un effet de flottement à l'écran. GCOL0,129 met la couleur du fond au rouge. Un ordre CLG, venant après, coloriera tout l'écran en rouge. Nous pouvons dans le même temps jouer l'air défini ci-dessus par l'appel de la procédure PROCmusique. L'information contenue dans PROCmusique est gérée plus rapidement que n'est jouée la partition. Aussi un tampon mémoire est-il utilisé pour sauvegarder l'information SOUND avant qu'elle ne soit exécutée.

**Niveaux de difficulté :** pour accentuer l'aspect défi du jeu, nous pouvons mettre en œuvre des niveaux de difficulté. Une fois le titre affiché, le programme demandera au joueur de donner un nombre de 0 à 9, qui sera sauvegardé comme variable-niveau de difficulté. Pour augmenter le nombre de mines, changez les lignes 1930 et 1940 de la sorte :

```
1930Niveau=difficulté+3+30
1940PROCminage(facteur)
```

En outre, lorsque nous repositionnerons les mines, nous devons calculer le nombre de mines qui restent en modifiant la ligne 3950 :

```
3950mines_restantes=facteur_score/150
```

Le listage complet de la procédure de page-titre est le suivant :

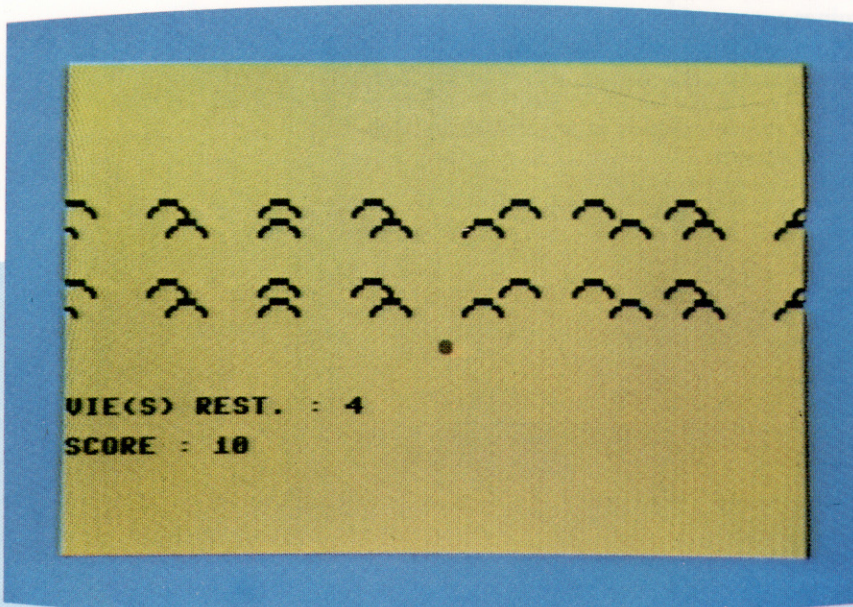
```
1300DEFpage_titre
1310GCOL 0,129
1320CLG
1330GCOL 3,3
1340PROCmusique
1350Y=100:Y=0
1360REPEAT
1370X=X+20:Y=Y+50
1380FOR I=1 TO 2
1390PROCmines
1400NEXT I
1410UNTIL Y>700
1420:
1430PROCmines
1440PRINTTAB(8,20)"Niveau de difficulté (0-9)?"
1450 PROCmusique
1460REPEAT
1470difficulté=GET-40
1480UNTIL difficulté<-1 AND difficulté<10
1490ENDPROC
1500:
1510DEF PROCmines
1520PLOT4,X,Y
1530REM ** LETTRE M **
1540PLOT1,0,200
1550PLOT1,80,-100
1560PLOT1,80,100
1570PLOT1,0,-200
1580REM ** LETTRE I **
1590PLOT0,40,0
1600PLOT1,80,0
1610PLOT0,-40,0
1620PLOT1,0,200
1630PLOT0,-40,0
1640PLOT1,80,0
1650REM ** LETTRE N **
1660PLOT0,40,-200
1670PLOT1,0,200
1680PLOT1,120,-200
1690PLOT1,0,200
1700REM ** LETTRE E **
1710PLOT0,160,0
1720PLOT1,-120,0
1730PLOT1,0,-200
1740PLOT1,120,0
1750PLOT0,-40,100
1760PLOT1,-80,0
1770REM ** LETTRE S **
1780PLOT0,200,50
1790PLOT1,0,40
1800PLOT1,-120,0
1810PLOT1,0,-100
1820PLOT1,120,0
1830PLOT1,0,-100
1840PLOT1,-120,0
1850PLOT1,0,40
1860ENDPROC
```

Nous avons jusque-là utilisé un programme d'appel provisoire qui réunit nos diverses procédures. Nous avons maintenant assemblé toutes les procédures nécessaires à la boucle principale du programme. Vous pouvez effacer le programme provisoire d'appel (les lignes 10 à 70) et saisir ce qui suit :

```
2020DEF PROCboucle
2030 REPEAT
2040PROCmise à jour_heure
2050PROCTest_clavier
2060aléatoire=RND(50-difficulté)
2070IF aléatoire=1 THEN PROCtireur-embusqué
2080 UNTIL TIME>12099 OR fin_drapeau=1
2090 ENDPROC
```

# Crabes

Le score, toujours le score. Possesseurs d'un ordinateur Commodore 64, n'hésitez pas : ce programme écrit en BASIC par Pierre Monsaut a de quoi vous plaire, et même en vacances !



Vous devez aider une pauvre crevette à regagner la mer en lui faisant éviter les crabes voraces qui patrouillent sur la plage. Chaque crevette amenée au but rapporte 10 points. Vous disposez de cinq vies pour tenter de marquer un score maximal. Utilisez les touches W pour avancer et Z pour reculer.

```

5 REM *****
10 REM * CRABES *
15 REM *****
20 GOTO 1000
50 PRINT N$;B$:
60 PRINT N$;A$:
70 PRINT BB$;BB$:
80 PRINT N$;B$:
90 PRINT N$;A$:
100 A$=RIGHT$(A$,1)+LEFT$(A$,39)
110 B$=RIGHT$(B$,39)+LEFT$(B$,1)
120 GET X$
130 P=P+40*((X$="W")-(X$="Z"))
140 IF P>PI THEN P=PI
150 IF P=PA THEN 2000
160 C=PEEK(P)
170 IF C<>32 AND C<>CP THEN 3000
180 POKE P1,CR
185 POKE P+N,9
190 POKE P,CP
200 P1=P
210 FOR I=1 TO 6
220 PRINT HH$:
230 NEXT I
240 T=T+1
250 IF T=500 THEN 3900
260 GOTO 50
1000 POKE 53280,6
1010 POKE 53281,7
1015 PRINT CHR$(147);
1020 X$=""
1030 A$=""
1040 B$=""
1045 N$=CHR$(144)
1046 N=54272
1050 PA=1244
1060 PI=1604
1070 P=PI
1080 P1=P
1090 CP=81
1100 CR=32
1110 HH$=CHR$(145)
1120 BB$=CHR$(17)
1150 T$=CHR$(19)
1160 RESTORE
1165 FOR I=1 TO 40
1170 READ A
1180 A$=A$+CHR$(A)
1190 NEXT I
1200 B$=A$
1260 NP=5
1265 T=0
1270 S=0
1280 X=INT(RND(TI)*37)+2
1290 A$=RIGHT$(A$,X)+LEFT$(A$,40-X)
1300 PRINT T$:
1310 FOR I=1 TO 17
1320 PRINT BB$:
1330 NEXT I
1340 PRINT N$;"VIE(S) REST. :";NP
1350 PRINT
1360 PRINT N$;"SCORE :";S
1380 FOR I=1 TO 13
1390 PRINT HH$:
1400 NEXT I
1410 GOTO 50
2000 S=S+10
2010 POKE P1,CR
2020 P=P1
2030 P1=P
2040 GOTO 1280
3000 POKE P1,CR
3020 P=P1
3030 P1=P
3100 U0=54296
3110 WA=54276
3120 AA=54277
3130 HF=54273
3140 LF=54272
3150 SS=54278
3160 PH=54275
3170 PL=54274
3180 POKE U0,15
3190 POKE WA,65
3200 POKE AA,190
3210 POKE PH,15
3220 POKE PL,15
3300 FOR I=1 TO 11
3310 READ H
3320 READ L
3340 READ DL
3350 GOSUB 3700
3360 NEXT I
3370 RESTORE
3380 FOR I=1 TO 40
3390 READ A
3400 NEXT I
3500 NP=NP-1
3510 IF NP=0 THEN 4000
3520 GET X$
3530 GOTO 1280
3700 POKE HF,H
3710 POKE LF,L
3720 POKE SS,136
3800 FOR J=1 TO DL
3810 NEXT J
3820 POKE HF,0
3830 POKE LF,0
3840 RETURN
3900 PRINT CHR$(147);
3910 FOR I=1 TO 12
3920 PRINT
3925 NEXT I
3930 PRINT TAB(12)"** TEMPS ECOULE **"
3940 FOR I=1 TO 1000
3950 NEXT I
4000 IF S>R THEN R=S
4010 NB=0
4020 PRINT CHR$(147);
4030 FOR I=1 TO 7
4040 PRINT
4050 NEXT I
4060 PRINT TAB(16);"SCORE :";
4070 PRINT S
4080 FOR I=1 TO 4
4090 PRINT
4100 NEXT I
4110 PRINT TAB(16)"RECORD :";
4120 PRINT R
4130 FOR I=1 TO 4
4140 PRINT
4150 GET X$
4160 NEXT I
4170 PRINT TAB(16);"UNE AUTRE ?"
4180 GET X$
4190 IF X$="" THEN 4180
4200 IF X$<>"N" THEN 20
4210 END
10000 DATA 117,184,105,32,32,117,184,105
10010 DATA 32,32,32,117,184,105,32,32
10020 DATA 117,184,105,32,32,32,32,32
10030 DATA 117,184,105,32,117,184,105
10040 DATA 32,32,117,184,105,32,32,32
20000 DATA 8,147,400,8,147,300
20010 DATA 8,147,100,8,147,400
20020 DATA 10,60,300,9,159,100
20030 DATA 9,159,300,8,147,100
20040 DATA 8,147,300,8,23,100
20050 DATA 8,147,400

```



# Apple attaque de nouveau

**Le Macintosh, avec son lecteur de disquettes et son moniteur intégrés, avec sa souris et son système d'exploitation, représente un important pas en avant dans la conception des ordinateurs.**

Le Macintosh diffère des autres ordinateurs que nous avons présentés jusqu'ici. Bien que le Macintosh soit d'abord une machine de gestion, Apple a décidé de créer son propre produit plutôt que de suivre la voie des autres constructeurs qui ont opté pour la comptabilité IBM. En prenant cette décision audacieuse, Apple a conservé sa réputation d'innovateur dans une industrie peuplée d'imitateurs.

La présentation du Macintosh est plutôt inhabituelle. L'unité système est de dimension réduite pour une machine offrant une telle puissance de traitement. L'affichage est produit sur un écran haute résolution de 9 pouces, et le lecteur de disquettes utilise des disquettes Sony de 3 1/2 pouces. Le boîtier comporte une poignée; le Macintosh peut donc être qualifié de machine véritablement portable. Le système complet, avec le clavier, la souris et un boîtier de transport optionnel, ne pèse que 11,600 kg. Le boîtier de transport est muni de compartiments destinés à tous les composants du Macintosh.

Le Macintosh possède un clavier mécanique offrant un excellent toucher, parfaitement adapté à une frappe rapide. Le clavier possède son propre processeur pour gérer des fonctions spéciales et des jeux de caractères internationaux. L'autre composant du Macintosh est la souris, ainsi nommée en raison de la « queue » qui la connecte à l'unité système. Ce dispositif a la taille d'un paquet de cigarettes. Il est déplacé sur une surface plane. Un curseur reproduit ses déplacements sur l'écran, et peut servir à sélectionner les opérations que l'on désire effectuer avec la machine. Cette approche est plus simple pour l'utilisateur que celle retenue par la plupart des machines où il est nécessaire de connaître une série de commandes.

## Petite souris

Par exemple, si vous désirez ouvrir un fichier document, vous n'avez qu'à manipuler la souris jusqu'à ce que le curseur se trouve au-dessus du petit pictogramme (icône) représentant une feuille de papier. Il suffit alors d'appuyer sur le bouton pour affecter l'écran à cette activité. Après avoir entré votre fichier au clavier, la souris vous permettra de revenir au menu principal; le fichier pourra alors être sauvegardé sur disquette en plaçant le curseur sur le symbole représentant une disquette.

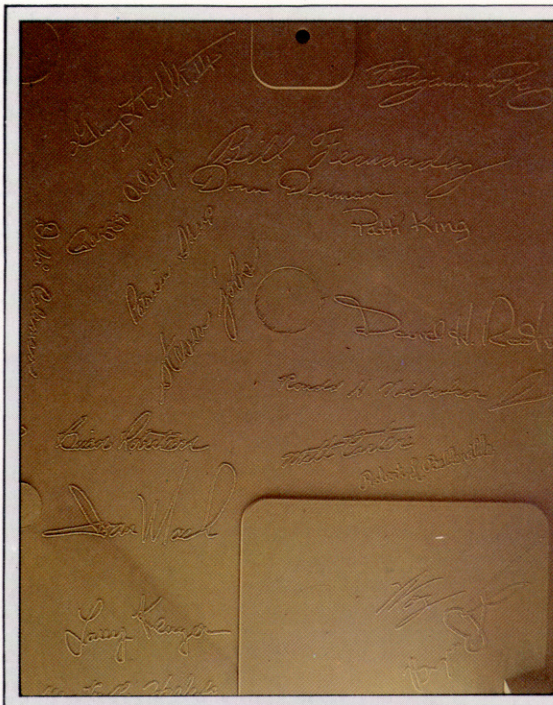
La version de base du Macintosh offre 128 K de mémoire utilisateur, laquelle peut être portée



à 512 K en remplaçant la RAM existante par des puces de 256 K. Le Macintosh a aussi 64 K de ROM renfermant le système d'exploitation qui gère presque toutes les opérations du système, ainsi que des fonctions spéciales. Le lecteur de disquettes utilise des disquettes qui stockent jusqu'à 400 K sur une face et qui sont plus fiables que les disquettes 5 1/4 pouces.

L'écran offre un affichage de 512 par 342 points, et il possède une topographie par bits

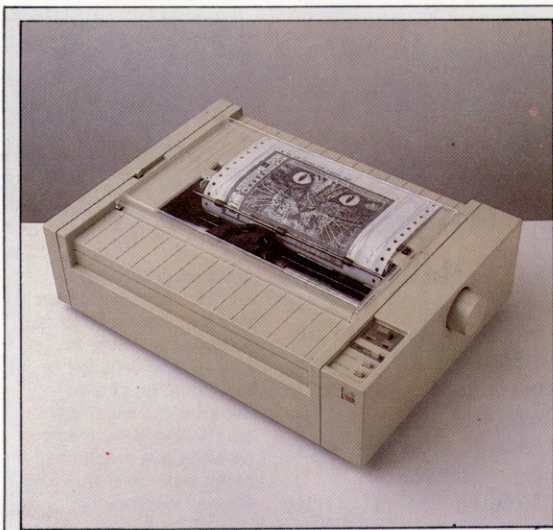
**Système Macintosh**  
Le Macintosh est conçu de façon à occuper le moins d'espace possible sur un bureau. L'excellente résolution de l'écran autorise des affichages graphiques généralement réservés aux machines coûtant dix fois plus cher. (Cl. Ian McKinnell.)



**Signatures**

A l'intérieur du boîtier du Macintosh, Apple a inscrit les signatures des membres de l'équipe de conception. On y retrouve, entre autres, celles de Steve Jobs et de Steve Wozniak, qui créèrent Apple. Leur génie rendit possible la création des ordinateurs Lisa et Macintosh.

grâce à laquelle chacun de ses 175 104 points peut être adressé individuellement. D'où des possibilités étonnantes au niveau des applications graphiques. En plus d'être très amusantes, les performances graphiques du Macintosh peuvent être très appréciées par des architectes ou par des photographes, par exemple. Comme il est conçu pour être utilisé avec l'imprimante rapide ImageWriter, tous ces graphiques peuvent être imprimés exactement comme ils apparaissent à l'écran. Malgré la grande qualité et la fiabilité du matériel Macintosh, c'est surtout son logiciel qui en fait une machine exceptionnelle. Avec l'intégration du matériel et du logiciel, et avec les commandes évoluées du système d'exploitation stockées en ROM, il est assez facile de transférer sur le Macintosh des programmes écrits pour d'autres machines. C'est relativement peu courant.



**Amélioration de l'image**

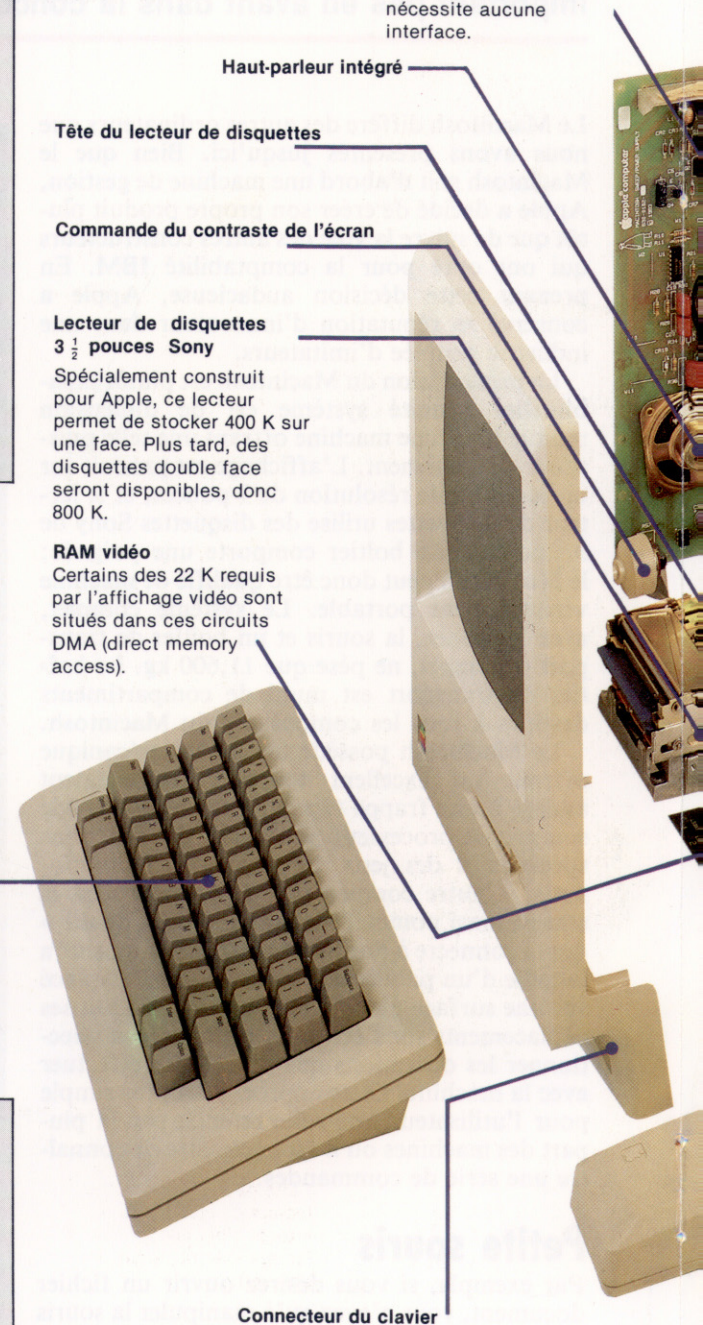
L'imprimante ImageWriter est une imprimante série qui produit un texte à une vitesse de 120 caractères par seconde. Sa vitesse est encore plus évidente en mode graphique, puisqu'elle génère les graphiques grâce à une topographie par bits.

**Clavier**

Le clavier détachable du Macintosh possède son propre processeur pour gérer les jeux de caractères étrangers et des fonctions spéciales. Les touches de curseur ne sont pas nécessaires en raison de la présence de la souris.

**Aide extérieure**

Le second lecteur de disquettes, conçu spécialement par Sony pour Apple, est un dispositif complémentaire appréciable pour le Macintosh. Sans lui, plusieurs opérations sont ralenties par la nécessité de changer de disquettes. Le lecteur externe ne nécessite aucune interface.



Haut-parleur intégré

Tête du lecteur de disquettes

Commande du contraste de l'écran

Lecteur de disquettes 3 1/2 pouces Sony

Spécialement construit pour Apple, ce lecteur permet de stocker 400 K sur une face. Plus tard, des disquettes double face seront disponibles, donc 800 K.

**RAM vidéo**

Certains des 22 K requis par l'affichage vidéo sont situés dans ces circuits DMA (direct memory access).

**Clavier**

Le clavier détachable du Macintosh possède son propre processeur pour gérer les jeux de caractères étrangers et des fonctions spéciales. Les touches de curseur ne sont pas nécessaires en raison de la présence de la souris.

Connecteur du clavier

**Souris**

La souris commande le mouvement du curseur et sert à « sélectionner » des objets à l'écran, puis gère l'activité correspondant à cet objet selon les instructions choisies dans un menu.



**Connexion analogique/numérique**  
Un câble-ruban connecte les cartes analogiques et numériques.

**Bus série**  
Nommées également « fentes virtuelles », les bus série vous permettent d'ajouter toute une variété de périphériques.

Sortie audio

Sortie d'imprimante

Connecteur de lecteur de disquettes externe

Connecteur de souris

## Macintosh d'Apple

### PRIX

\*\*\*\*

### DIMENSIONS

343 x 254 x 254 mm (boîtier écran/lecteur de disquettes).

### UC

Motorola 68000.

### MÉMOIRE

RAM DE 128 K, ROM de 64 K.

### ÉCRAN

Moniteur monochrome intégré, 512 x 342 points, fenêtres, menus, icônes.

### INTERFACES

Souris, imprimante, lecteur de disquettes externe, amplificateur hi-fi, bus série.

### LANGAGES DISPONIBLES

Basic, Cobol, Pascal.

### CLAVIER

De type mécanique, 59 touches, clavier numérique optionnel.

### DOCUMENTATION

Un manuel d'utilisation accompagné d'une cassette audio et d'une disquette d'introduction. Des manuels sont fournis avec MacPaint et MacWrite, qui sont aussi accompagnés d'une disquette et d'une cassette.

### FORCES

Le Macintosh dispose d'un logiciel très puissant et facile à utiliser. La souris simplifie l'utilisation de la machine. La conception technique facilite l'entretien et la mise à jour.

### FAIBLESSES

Le Macintosh est un micro qui n'est pas à la portée de nombreux utilisateurs. Le lecteur de disques unique ralentit et complique les opérations sur disquettes. Peu de programmes sont offerts actuellement.

**Contrôleur de communications série**

**Contrôleur de disquettes**  
Cette puce commande le lecteur de disquettes intégré ainsi que le lecteur externe, lorsqu'il est ajouté.

**UC**  
Motorola 68000 : cette puce effectue des opérations internes 32 bits à la fois, mais envoie et reçoit des données de 16 bits.

**RAM utilisateur de 128 K**  
Ces 16 puces peuvent être remplacées par des puces RAM de 256 K, ce qui donnerait au Macintosh une mémoire utilisateur totale de 512 K.

**Adaptateur d'E/S 6522**

Le 6522, aussi utilisé pour la commande d'E/S de l'Apple IIe, gère le clavier, la souris et les circuits d'horloge à temps réel.

**Carte analogique**

Cette carte commande le moniteur vidéo et l'alimentation. Le Macintosh n'a pas besoin de ventilateur. L'excès de chaleur est diffusé par des plaques métalliques situées près des ouvertures de ventilation du boîtier.





Apple a investi massivement dans la technologie du Lisa et du Macintosh. Sur le plan de la qualité de la conception et de la construction, l'introduction du Macintosh est véritablement un événement.

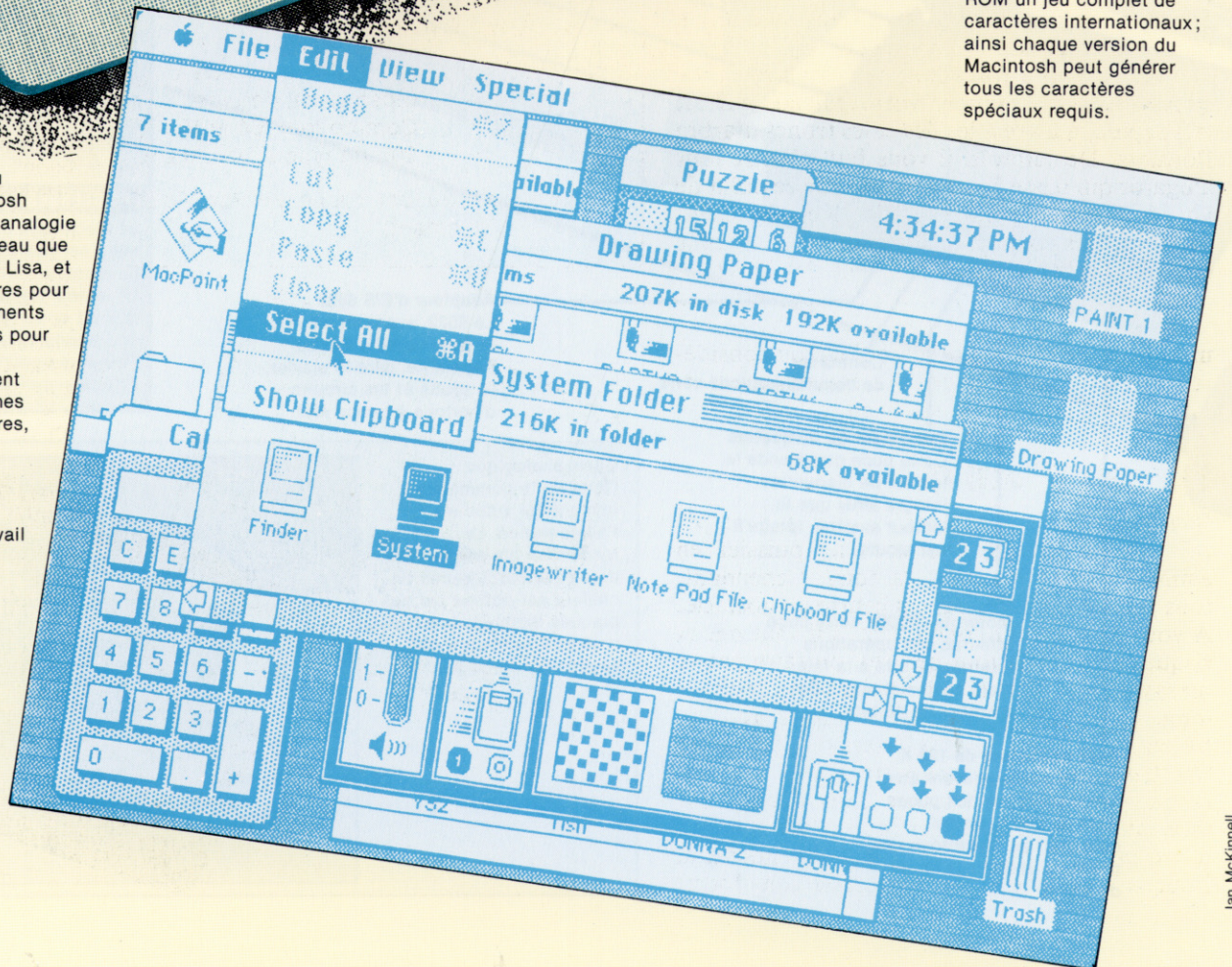


### Clavier

Les premières livraisons de Macintosh comportaient un clavier américain. Le clavier du Macintosh possède en ROM un jeu complet de caractères internationaux; ainsi chaque version du Macintosh peut générer tous les caractères spéciaux requis.

### Matériel de bureau

L'écran du Macintosh présente la même analogie au matériel de bureau que celle créée pour le Lisa, et il utilise des fenêtres pour afficher des documents qui sont trop longs pour entrer sur l'écran. Les fenêtres peuvent être placées les unes au-dessus des autres, de telle sorte que l'utilisation du Macintosh se rapproche beaucoup d'un travail traditionnel sur un bureau.





# Sauvé des eaux

**Dans River Rescue, il vous faudra venir en aide à des savants en détresse, tout en affrontant des ennemis. Ce jeu est désormais disponible sur plusieurs micro-ordinateurs.**

River Rescue est avant tout un jeu d'arcades, où il faut tirer sur tout ce qui bouge : il n'a pas d'autres prétentions. On le doit à Creative Sparks, la filiale « logiciels » de Thorn EMI. Le soutien d'une aussi grosse compagnie a permis la sortie de quatre versions différentes : pour le Spectrum 48 K, le Commodore 64, le Vic-20 (sans extension mémoire) et les ordinateurs Atari. Il est fourni dans une enveloppe en plastique, et non dans l'habituelle boîte transparente. Le tout est accompagné d'une notice très bien conçue dans la mesure où sa lisibilité est suffisamment claire pour être d'un précieux secours. Car, dans de tels jeux d'actions, savoir bien manipuler ne garantit pas le succès. Il faut aussi connaître un certain nombre de « trucs », sinon le découragement risque d'arriver.

Le jeu en lui-même est très simple, mais bourré d'actions. Vous contrôlez un bateau à moteur chargé de venir en aide à une équipe de savants bloqués en amont de la rivière. On ne sait d'ailleurs pas exactement ce qui leur est arrivé, mais comme vous avez pour mission de les conduire à l'hôpital, on peut craindre le pire.

Tout en vous efforçant de sauver les scientifiques, vous devez guider votre bateau, qui est très rapide, à travers les îles et les troncs d'arbre flottants. De surcroît, il vous faut abattre tout alligator qui passe à votre portée. De ce point de vue, la version destinée au Vic-20 est agrémentée de canoës coulés et d'anacondas. A certains endroits de la rivière se trouvent des jetées en bois ; c'est là que vous devrez récupérer les savants. Chaque fois que vous en faites passer un sur l'autre rive, votre score s'accroît considérablement ; mais vous pouvez également exterminer les alligators qui infestent les lieux.

## Pur jeu d'action

Il est plus avantageux de transférer les rescapés par groupe (encore que vous ne puissiez en emmener plus de neuf à la fois). Cependant, tous périront si votre bateau heurte un obstacle. A vous de voir, par conséquent, s'il vaut mieux risquer gros (mais tout perdre), ou faire passer un savant à la fois, ce qui est plus sûr, mais plus long. Car il vous faut aussi compter avec un hélicoptère (un avion dans la version destinée au Spectrum) qui peut apparaître à tout moment et qui peut jeter des mines dans l'eau ; vous serez alors contraint de les faire exploser avant d'entreprendre quoi que ce soit de nouveau. Pour le Vic-20, le jeu est disponible sous forme

de cartouche, et vous n'aurez donc pas à attendre que le chargement — toujours très long sur cassette — ait pris fin. Cette version vous propose deux options — trois ou six savants — et vous accorde six vies. Les points acquis au cours de chacune d'elles viennent s'ajouter à votre score ultérieur : les choses en sont facilitées d'autant. Il est vrai que vous devrez remonter trois rivières supplémentaires...

River Rescue est un pur jeu d'action et de réflexe, très soigneusement conçu, et d'une difficulté suffisante pour vous occuper un bon moment ; mais peut-être lui aurait-il fallu ce petit rien qui fait les très grands jeux... et les cassettes d'or !

**River Rescue :** Spectrum 48 K; Atari; Commodore 64; Vic-20 (sans extension mémoire).

**Éditeurs :** Creative Sparks.

**Auteur :** Kevin Buckner.

**Manches à balai :** Kempston/Interface 1 (Spectrum); de type Commodore (64 et Vic-20); de type Atari (Atari).

**Format :** cassette; cartouche (version Vic-20 uniquement).

### A la rescousse

On voit ici la version Spectrum de River Rescue. Le premier cliché montre le générique, qui donne une « bonne » idée des graphismes du jeu. Le second reproduit une phase du jeu. Le bateau transporte un savant et le mène vers les autres, déjà tirés d'affaire, là où il sera en sécurité. (Cl. Liz Heaney.)





# La récursivité

La récursivité peut donner une nouvelle dimension à vos programmes en BASIC. Nous vous exposons ici l'utilisation pratique de cette technique sur un exemple simple, le jeu des « Tours d'Hanoi ».

L'objet de cette recherche peut être illustré par une définition sous forme de boutade : « Récursivité : voir *Récursivité*. »

Cette définition qui « tourne en rond » illustre bien le principe même de la récursivité — quelque chose qui se définit par soi-même. Il manque cependant une condition indispensable à la viabilité de la récursivité : qu'il y ait une issue à ce cercle vicieux.

Pour illustrer le principe, nous avons pris l'exemple de l'énigme dite « les Tours d'Hanoi ». Il s'agit d'une pile de disques dont la taille va en décroissant vers le haut. L'énigme consiste

à reconstituer une autre pile selon les règles suivantes :

1. Un seul disque doit être déplacé à la fois.
2. On ne peut placer un disque sur un autre de taille inférieure.
3. On ne peut constituer plus de trois piles à la fois.

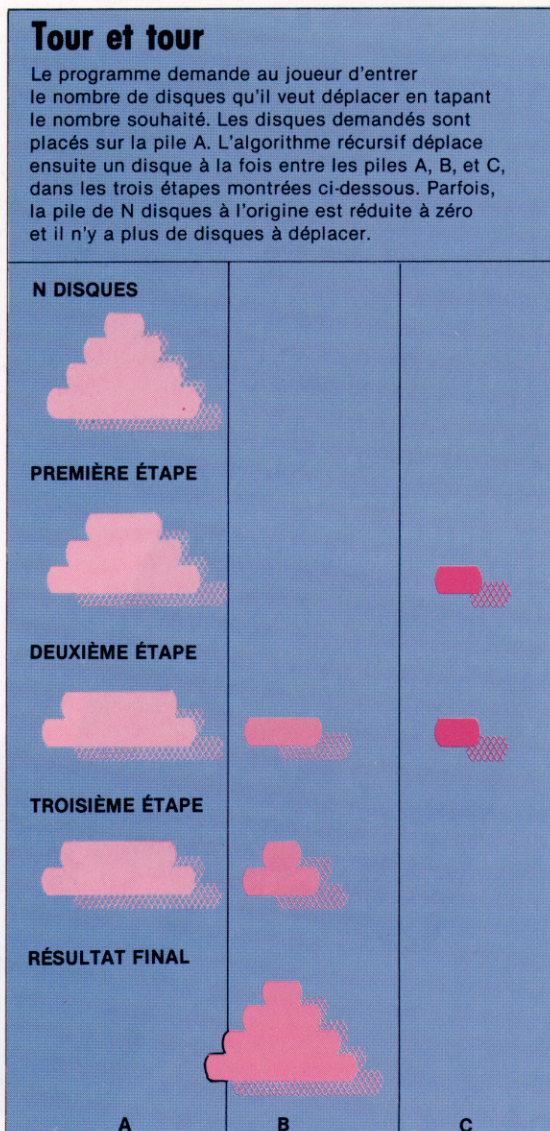
Le schéma ci-contre montre comment le concept de récursivité peut résoudre le problème. Nous partons d'une pile de quatre disques. Une variable N indiquera le nombre total de disques à déplacer et recevra ici la valeur 4. Puisque les règles du jeu n'autorisent pas à déplacer plus d'un disque à la fois, nous utiliserons une formule récursive pour réduire N de 1, et nous poursuivrons le calcul jusqu'à ce que N soit égal à 1. Lorsque N = 1, le programme cesse les calculs et déplace le disque approprié.

Si nous travaillons avec une version du BASIC qui permet la récursivité, il sera facile d'écrire un programme qui suive exactement cette démarche. Pour le programme BASIC du micro-ordinateur BBC, le travail de calcul des déplacements a lieu entre les lignes 1000 et 1050. Le reste du programme sert à afficher les déplacements successifs.

Pour transcrire le programme BASIC BBC des Tours d'Hanoi, en BASIC Spectrum, il nous faut remplacer une procédure récursive par un sous-programme récursif à la ligne 1000 de notre listing. A chaque appel récursif par le sous-programme des tableaux M, A, B ou C, la variable pointeur J est incrémentée, et les nouvelles valeurs pour la variable sont mises dans M(J), A(J), B(J) et C(J). Ces nouvelles valeurs pourront ensuite être utilisées dans le prochain appel du sous-programme sans affecter les anciennes. A la fin du sous-programme, la valeur de J est décrémentée pour restituer les valeurs d'origine. Cette méthode peut toujours être utilisée pour écrire des sous-programmes récursifs en BASIC.

La partie du programme consacrée à l'affichage des déplacements est simple.

Lors de chaque déplacement, la nouvelle position du disque est affichée, et l'ancienne est effacée par des caractères vides. Les programmes montrent les disques selon une vue verticale. Afin de produire un effet de symétrie sur les piles, nous avons ombré chaque disque par des caractères graphiques à une seule de ses extrémités, caractères constitués pour moitié d'un espace et pour moitié d'une couleur pleine.



Liz Dixon



## BBC Micro

```

10 DIM M(10): DIM A(10): DIM B(10): DIM C(10)
20 DIM D$(10,10): DIM H(3): DIM P(3,10)
30 GO SUB 3000
90 DIM M(100): DIM A(100): DIM B(100):
   DIM C(100)
100 INPUT "NOMBRE DE DISQUES? " :N
110 IF N<1 OR N>10 THEN GO TO 100
120 GO SUB 3100
130 LET J=1: LET M(J)=N: LET A(J)=1: LET B(J)
   =2: LET C(J)=3
140 GO SUB 1000
200 STOP
1000 IF M(J)=1 THEN GO SUB 1500: RETURN
1010 LET J=J+1
1020 LET M(J)=M(J-1)-1
1030 LET A(J)=A(J-1)
1040 LET B(J)=C(J-1)
1050 LET C(J)=B(J-1)
1060 GO SUB 1000
1100 LET M(J)=1
1110 LET A(J)=A(J-1)
1120 LET B(J)=B(J-1)
1130 LET C(J)=C(J-1)
1140 GO SUB 1000
1200 LET M(J)=M(J-1)-1
1210 LET A(J)=C(J-1)
1220 LET B(J)=B(J-1)
1230 LET C(J)=A(J-1)
1240 GO SUB 1000
1300 LET J=J-1
1310 RETURN
1500 LET PA=A(J): LET PB=B(J)
1510 LET M#=D$(P(PA,N+1-H(PA)))
1520 FOR I=22-H(PA) TO 7 STEP -1
1530 PRINT AT I-1,10*(PA-1);M#;
1540 PRINT AT I,10*(PA-1);B#;
1550 NEXT I
1560 FOR I=10*(PA-1) TO 10*(PB-1) STEP SGN (PB-PA)
1570 PRINT AT 6,I;M#;
1575 PRINT AT 6,I;B#;
1580 NEXT I
1590 FOR I=6 TO 20-H(PB)
1600 PRINT AT I,10*(PB-1);B#;
1610 PRINT AT I+1,10*(PB-1);M#;
1620 NEXT I
1640 LET H(PB)=H(PB)+1: LET P(PB,N+1-H(PB))=P
   (PA,N+1-H(PA))
1650 LET P(PA,N+1-H(PA))=0: LET H(PA)=H(PA)-1
1660 RETURN
3000 LET B$="      ": LET C$=CHR$ 143+CHR$
   143+CHR$ 143+CHR$ 143
3010 LET C$="": FOR I=1 TO 10: LET C$=C$+CHR$
   143: NEXT I
3020 FOR I=1 TO 9 STEP 2
3030 LET D$(I)=B$( TO 4-INT (I/2))+CHR$ 133+C$
   (TO 2*INT (I/2))+CHR$ 138+B$( TO 4-INT (I/2))
3040 LET D$(I+1)=B$( TO 4-INT (I/2))+C$( TO
   I+1)+B$( TO 4-INT (I/2))
3050 NEXT I
3060 RETURN
3100 ENCRE 3: PAPIER 6: MARGE 6: CLS
3120 FOR I=1 TO N
3130 PRINT AT 21-N+I,0;D$(I);
3135 LET P(1,I)=I: LET P(2,I)=0: LET P(3,I)=0
3140 NEXT I
3150 LET H(1)=N: LET H(2)=0: LET H(3)=0
3160 RETURN

```

## Spectrum

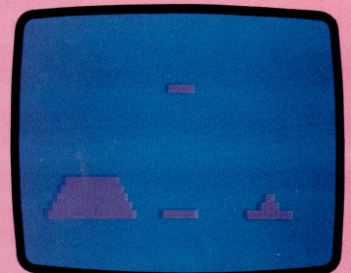
```

10 DIM D$(12),H(3),P(3,12)
20 PROCINIT
100 INPUT "NOMBRE DE DISQUES (1-12) " :N
110 IF N<1 OR N>12 THEN 100
120 PROCFAFFICHAGE(N)
130 PROCHANOI(N,1,2,3)
200 END
1000 DEFPROCHANOI(M,PA,PB,PC)
1010 IF M=1 THEN PROCMOVE(PA,PB):
   ENDPROC
1020 PROCHANOI(M-1,PA,PC,PB)
1030 PROCHANOI(1,PA,PB,PC)
1040 PROCHANOI(M-1,PC,PB,PA)
1050 ENDPROC
1100 DEFPROCDEPLACEMENT(PA,PB)
1110 D$=D$(P(PA,N+1-H(PA)))
1120 FOR I=24-H(PA) TO 10 STEP -1
1130 PRINT TAB(13*(PA-1),I);B#;
1140 PRINT TAB(13*(PA-1),I-1);D#;
1150 NEXT I
1160 FOR I=13*(PA-1) TO 13*(PB-1)
   (PB-PA)
1170 PRINT TAB(I,9);D#;
1180 NEXT I
1190 FOR I=9 TO 22-H(PB)
1200 PRINT TAB(13*(PB-1),I);B#;
1210 PRINT TAB(13*(PB-1),I+1);D#;
1220 NEXT I
1240 H(PB)=H(PB)+1: P(PB,N+1-H(PB))
   =P(PA,N+1-H(PA))
1250 P(PA,N+1-H(PA))=0: H(PA)=H(PA)-1
1260 ENDPROC
3000 DEFPROCINIT
3020 FOR IX=1 TO 11 STEP 2
3030 D$(IX)=CHR$150+STRING$(5-
   IX/DIV2," ") +CHR$234+STRING$(2*
   (I+DIV2),CHR$255)+CHR$53+STRING$(
   5-IX/DIV2," ")
3040 D$(IX+1)=CHR$150+STRING$(5-
   IX/DIV2," ") +STRING$(IX+1,CHR$
   255)+STRING$(5-IX/DIV2," ")
3050 NEXT IX
3060 B$=CHR$150+STRING$(12," ")
3070 VDU 23,1,u;0;0;0;
3080 ENDPROC
3100 DEFPROCFAFFICHAGE (N)
3110 CLS
3120 FOR I=1 TO N
3130 PRINT TAB(0,23-N+I);D$(I);
3135 P(1,I)=I:P(2,I)=0:P(3,I)=0
3140 NEXT I
3150 H(1)=N:H(2)=0:H(3)=0
3160 ENDPROC

```

### Problèmes récurrents

Voici une photographie du programme des Tours d'Hanoi fonctionnant sur un Spectrum. Les couleurs des blocs peuvent être changées. Si vous essayez de suivre la manière dont l'ordinateur résout le problème, attention, les déplacements se font très vite !





# La routine

**Nous avons vu comment un programme peut être décomposé en petites unités appelées modules. Nous abordons aujourd'hui leur intégration dans le développement d'un programme complet.**

Lorsque l'on développe un programme, il est conseillé de mettre au point une structure globale constituée d'un ensemble clé de routines d'ordre général qui seront utilisées ultérieurement par des routines de plus en plus spécialisées. L'ensemble est placé sous le contrôle d'un seul module de commande, au sommet. Cette structure permet de distinguer des niveaux de programmation de plus en plus précis, créant un effet de pyramide.

Au fur et à mesure que l'on progresse dans l'organisation ainsi hiérarchisée, on atteint des niveaux de plus en plus élémentaires dans la pyramide, jusqu'à la base. Il faut donc commencer par le programme de commande. Ses fonctions sont définies en termes d'appels de routines de niveaux inférieurs. Nous ne nous y intéresserons pas ici. Après le programme de commande, nous passons au niveau immédiatement inférieur, celui des routines principales. Ces dernières sont définies selon les autres routines qu'elles doivent appeler. Et l'on répète ce processus d'appels à des routines de niveaux de plus en plus élémentaires.

Prenons comme exemple le développement d'un programme de jeu, « le jeu du pendu ». Au lieu que le joueur essaie de deviner un mot choisi par le programme, comme c'est le cas pour la plupart des versions de programme propres à un ordinateur, nous voulons, à l'inverse, que le programme devine un mot que nous choisissons. Parmi les manières de traiter ce problème, et sans avoir à donner à l'ordinateur une longue liste de mots, nous pouvons saisir des données ayant trait à la probabilité d'occurrence de certaines séquences de caractères.

```
100 REM Initialisation des variables et des tableaux
/
/
/
500 REM *****Routine de commande*****
510 REM
520 GOSUB 1000:REM Pages-écran de titre et d'assistance
530 GOSUB 2000:REM constituer la page
540 GOSUB 4000:REM Trouver la longueur du mot du joueur
550 GOSUB 8000:REM Choisir le jeu de données et le charger
560 GOSUB 3000:REM Deviner une lettre
570 GOSUB 4500:REM Tentative avec le joueur de deviner le mot
580 GOSUB 5000:REM Mettre à jour la page
590 IF JEU_NON_TERMINE THEN 560:REM essayer à nouveau de deviner-jusqu'à la fin du jeu
```

```
600 IF GAGNE THEN GOSUB 10000 ELSE GOSUB
610 GOSUB 6000:REM donner la fin appropriée du mot - une seule tentative
620 GOSUB 6000:REM demander un autre jeu au joueur
630 IF AUTRE_JEU THEN 530:REM recommencer si autre jeu
640 GOSUB 7000:REM dire au revoir et terminer
650 END
```

Avant de commencer, il est nécessaire d'initialiser les variables, dimensionner les tableaux, constituer l'affichage de la page, et éventuellement de le mettre à jour. Il faut aussi écrire des routines pour la tenue du score, pour les tentatives de deviner le mot, et pour la fin du jeu.

Notre approche de l'écriture de la routine de commande est une simple instruction REM qui permet d'indiquer que les variables et les tableaux doivent être dimensionnés. La routine de commande est constituée de deux boucles. La boucle externe (ligne 620) effectue un test pour détecter si l'utilisateur indique la fin du jeu, la boucle interne (ligne 590) exécute un test pour déterminer si le jeu est fini.

Si nous voulons pouvoir tester la routine de commande, il nous faut mettre en place les sous-programmes fictifs pour répondre aux instructions GOSUB. Chaque GOSUB de la routine de commande doit se voir associer une instruction REM qui explique sa fonction. Ces commentaires doivent se situer de préférence à un numéro de ligne facile à retenir ou à repérer, un chiffre rond, tel que 1000 ou 5000. Il convient également de vérifier que des routines assurant des fonctions similaires reçoivent toujours des numéros de lignes standards. Cela facilite le transfert des routines d'un programme à l'autre. Par exemple, des instructions de jeu commenceront à la ligne 1000 et une ligne de programme GOSUB 7000 terminera toujours un jeu en appelant une routine standard.

Notre routine initiale de commande reste simple et courte. Elle figure sur une page-écran et elle est donc plus facile à comprendre et à mettre au point (dépistage d'erreurs) qu'un programme sur plusieurs pages-écrans. Les trois variables, JEU\_NON\_TERMINE, GAGNE et AUTRE\_JEU, sont des drapeaux dont l'état est affecté dans les différents sous-programmes appelés par la routine de commande. Ils sont utilisés ici pour déterminer si le programme de commande fonctionne de manière satisfaisante. Il est relativement facile en effet de localiser une erreur de

logique dans ce sous-programme simple de commande.

Il est maintenant nécessaire d'examiner de plus près la structure du programme. Il faut s'assurer notamment que le programme fonctionne normalement dans toutes les circonstances. Nous pouvons également commencer à apporter des améliorations à sa structure. Nous pourrions, par exemple, rendre accessibles à n'importe quel moment du jeu les instructions. Ce pourrait être également une bonne idée de comptabiliser le nombre de jeux gagnés par l'ordinateur et par le joueur.

L'étape suivante consiste à préciser les sous-programmes appelés par le programme de commande. Les listages que nous donnons ici montrent le fonctionnement de ces routines. La première (elle commence à la ligne 4000) demande seulement à l'utilisateur un nombre entre 0 et 20 (la longueur du mot). Elle utilise un sous-programme d'utilité générale présent à la ligne 51000. Il ira chercher une chaîne spécifiée dans PROMPT\$, l'affichera avant de recevoir la saisie d'un nombre au clavier. Si ce n'est pas une valeur entière comprise entre MIN% et MAX%, un message d'erreur est signalé et l'utilisateur se voit demandé de saisir un autre nombre. Ce sous-programme peut être facilement mis en œuvre dans d'autres programmes, et une librairie de modules d'utilité générale pourra être constituée en vue de projets ultérieurs.

```

4000 REM Trouver la longueur du mot spécifié
    par le joueur
4010 REM
4020 PROMPT$="Combien y-a-t'il de lettres
    dans votre mot ?"
4030 MINX=1
4040 MAXX=20
4050 GOSUB 51000:REM Tapez un nombre entier
    compris entre MINX et MAXX
4060 LONGMOTX=REX:REM REPX est utilisé par
    le sous-programme en 51000 pour communiquer
    la réponse
4070 RETURN
    
```

```

8000 REM choisir le jeu de données et le
    charger
8010 REM
8020 IF LONGMOTX>7 THEN LONG_FIX=8
    ELSE LONG_FIX=LONGMOTX
8030 FILEND_L$=CH$(LONG_FIX)
8040 NOMFI$="TABLE"+FILEND_L$
8050 GOSUB 9000:REM OPEN, READ & CLOSE
    (ouvrir, lire et fermer le fichier), avec les
    données probables pour la bonne longueur de
    mot
8060 RETURN
    
```

Puisque nous n'acceptons qu'un seul jeu de données (une table) à la fois en RAM, il nous faut établir une chaîne de caractères dans NOMFI\$ pour recevoir le nom du fichier de données. Il nous faudra ensuite appeler le sous-programme à la ligne 9000 pour lire le fichier.

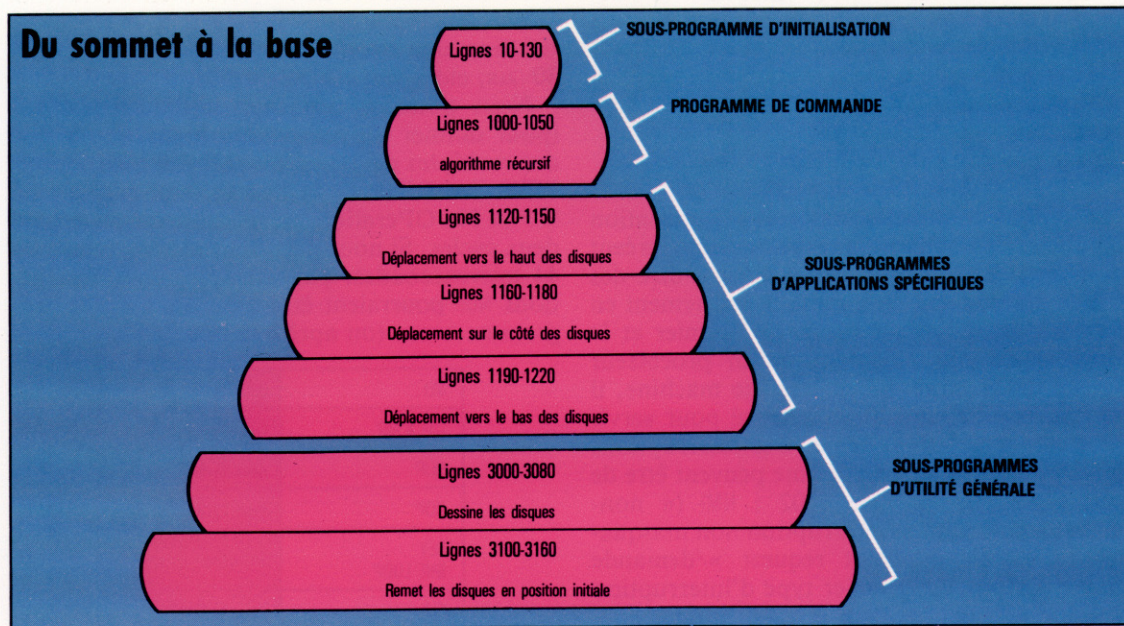
Nous verrons que notre programme va souvent directement d'une routine à l'autre. Nous aurons pourtant probablement besoin de créer une routine supplémentaire qui appellera alternativement chacune des deux routines.

Cette utilisation de sous-programmes portables d'un programme à l'autre suppose un travail supplémentaire, et il faut accorder un grand soin à la création de routines applicables à de nombreuses situations très variées. On peut y parvenir simplement en remplaçant les constantes par des variables. Il est essentiel, par ailleurs, que tous les sous-programmes soient parfaitement documentés. Il faut notamment spécifier la destination précise du sous-programme, donner des détails sur les variables utilisées, préciser les valeurs attendues en saisie et en sortie de traitement, et tous les effets annexes du sous-programme.

Une présentation standard des sous-programmes est très utile; assurez-vous que les numéros de lignes sont d'un pas déterminé (avancent de 10 en 10 par exemple), les titres et les commentaires étant limités à un nombre donné de lignes au début de la routine, RETURN figurant toujours sur la dernière ligne.

**Programmation en pyramide**

Ce schéma représente la programmation hiérarchisée ou en pyramide. Nous avons utilisé le programme des Tours d'Hanoi de la page 955. Les numéros de lignes qui figurent sur ce schéma renvoient au listage pour le BBC Micro. La première couche de cet édifice représente le programme d'initialisation qui doit être complet pour que le programme puisse être exécuté. Le programme de commande, qui vient après, représente l'algorithme récursif qui effectue les calculs et les appels nécessaires aux sous-programmes. Viennent ensuite les sous-programmes d'applications spécifiques (des lignes 1120 à 1220). Ils servent à déplacer d'une pile à l'autre les disques des tours d'Hanoi pour l'affichage. Enfin, la base de la pyramide est constituée des sous-programmes d'utilité générale. Ils servent à formater l'affichage initial et à dessiner les disques des tours. En comparant cette structure au listage, vous verrez que le programme suit exactement la même séquence.



LIZ DIXON

# Pas si vite

**Les programmeurs en langage d'assemblage trouvent souvent que leurs programmes vont trop vite. Voici les méthodes les plus courantes pour créer des retards dans les logiciels 6502 et Z80.**

Différents moyens permettent de réaliser des boucles de retard en langage d'assemblage 6502. La méthode la plus évidente consiste à charger l'un des registres d'index avec une valeur et à la décrémenter dans une boucle jusqu'à zéro :

BOUCLE DE RETARD	TEMPS PRIS POUR CHAQUE OPÉRATION
LDY # \$07	Deux cycles
DEY	Deux cycles
BNE LOOP	Deux cycles (3 cycles si branchement sur la même page; 4 cycles pour page différente).

cher ces interruptions, mais il est possible d'arrêter les interruptions IRQ qui ne sont pas vitales pour le fonctionnement du processeur.

Les IRQ peuvent être masquées en mettant à 1 un bit particulier dans le PSR (registre d'état du processeur). C'est l'instruction SEI qui fait cela. IRQ peut être réactivé en remettant à zéro le même bit à l'aide de CLI. Si nous masquons IRQ avant d'entrer la boucle de retard, nous pouvons améliorer sa précision. Si une interruption non masquable a lieu au cours de l'exécution, elle causera des erreurs de synchronisation. Notre listage initial de boucle de retard devra être ainsi modifié pour masquer les interruptions :

INSTRUCTION	FONCTION	TEMPS PRIS
SEI	Désactive IRQ	Deux cycles
LDY # \$07		} 36 cycles
DEY		
BNE LOOP		
CLI	Réactive IRQ	Deux cycles

Chaque instruction en langage machine prend un certain nombre de cycles d'horloge pour être exécutée. Par exemple, l'instruction DEY prend deux cycles et LDY en adressage immédiat en prend aussi deux. Comme chaque cycle dure une microseconde ( $\mu s$ ), nous pouvons calculer le « temps réel » pris pour exécuter la boucle de retard. Le total des cycles peut être calculé comme suit :

1. Instruction LDY # \$07 : deux cycles.
  2. Sept branchements du programme. A chaque branchement, l'opération BNE prend trois cycles : donc les instructions DEY et BNE prennent  $(2 + 3) \times 7 = 35$  cycles.
  3. Mais le dernier BNE ne se branche pas et il ne prend donc que deux cycles.
- D'où le nombre total de cycles :
- $$2 + 35 - 1 = 36.$$

Différents problèmes, associés à l'utilisation de boucles de retard en langage machine, créent des retards en « temps réel » (c'est-à-dire des retards qui peuvent être mesurés exactement en secondes ou microsecondes). Le premier et le principal est que, pendant qu'un processeur exécute un programme en langage machine, il suspend régulièrement son activité pour servir d'autres parties du système. Ces interruptions dans l'exécution du programme peuvent être de deux types sur la puce 6502 : NMI (« non-maskable interrupt »/interruption non masquable) et IRQ (« interrupt request »/demande d'interruption). Le premier type d'interruption implique que rien ne peut être fait pour empê-

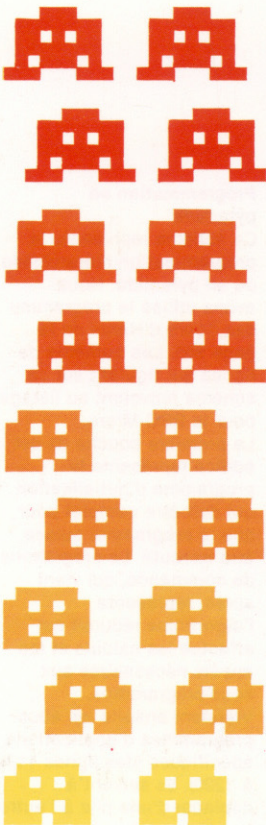
En masquant de la sorte les IRQ, on ajoute quatre autres cycles à la routine, ce qui a pour effet un retard total de  $40 \mu s$ , en supposant qu'il n'y ait pas de NMI.

Un autre aspect des boucles de retard est la « résolution » — c'est-à-dire la variation du temps pris pour exécuter une boucle de retard — entre une valeur de compteur et la suivante. Dans notre exemple, nous chargeons le registre Y avec 7, mais si nous avons utilisé 6 à la place, le retard aurait été de :

$2 + 2 + (2 + 3) \times 6 - 1 + 2 = 35 \mu s$ .  
Et avec une valeur de 5, nous aurions pris  $30 \mu s$ , etc.

Nous pouvons peaufiner notre programme (pour donner des synchronisations autres que des multiples de 5), en plaçant des instructions NOP à l'extérieur de la boucle. Une instruction NOP (« No OPERATION »/pas d'opération) prend deux cycles. Si nous désirons créer un retard de 44 microsecondes, par exemple, deux instructions NOP pourraient être ajoutées à notre programme avant (ou après) la boucle :

SEI	deux cycles
LDY # \$07	deux cycles
NOP	deux cycles
NOP	deux cycles
DEY BNE LOOP	34 cycles
CLI	deux cycles





Ce type de retard a une limite supérieure de temps déterminée par la valeur maximale de Y pouvant être utilisée. Comme le registre Y est sur huit bits, ce maximum est 255. D'où :

$$2 + 2 + (2 + 3) \times 255 - 1 + 2 = 1\,280 \mu s,$$

soit à peu près un millième de seconde. C'est très long en termes de microprocesseur, mais pas pour les humains. Il nous arrivera d'avoir besoin de retards plus importants. De petites améliorations dans la longueur de temps peuvent être apportées en ajoutant des instructions NOP à l'intérieur de la boucle. Ainsi, en ajoutant un NOP, on augmente le retard maximal de :

$$2 + 2 + (2 + 2 + 3) \times 255 - 1 + 2 = 1\,790 \mu s.$$

Pour des retards plus importants, il nous faut imaginer d'autres méthodes. Les deux les plus courantes consistent soit à utiliser une seconde boucle emboîtée autour de la première, soit à décrémenter un nombre plus grand, tel qu'un mot de 16 bits consistant en deux octets de mémoire. Pour chacune, on peut calculer le standard de résolution pouvant être obtenu.

```

DELAI SEI
      LDX ##04          ;ut. reg X comme compteur
de boucle ext.
LOOP1 LDY ##FF        ;ut. reg Y comme compteur
de boucle int.
LOOP2 DEY
      BNE LOOP2        ;fin de boucle int.
      DEX
      BNE LOOP1        ;fin de boucle ext.
      CLI
  
```

La boucle interne du programme ci-dessus prend  $2 + (2 + 3) \times 255 - 1 = 1\,276 \mu s$  à être exécutée. La boucle externe contrôle l'exécution de la boucle interne et exécute quatre fois DEX et BNE. Le total de ce retard est :

$$2 + 2 + (1\,276 + 2 + 3) \times 4 - 1 + 2 = 5\,129 \mu s.$$

## Retards sur Z80

Chaque instruction en langage machine Z80 prend un temps différent (mesuré en unités appelées « état T »), et la vitesse du Z80 diffère suivant les machines. Pour calculer le temps réel pris par chaque instruction, on divise le nombre d'états T pour l'instruction par la fréquence d'horloge du micro. Ainsi, une instruction qui prend quatre états T sur un processeur avec une fréquence d'horloge de 2 MHz est exécutée en deux microsecondes.

L'ouvrage de Rodnay Zaks, *Programmation du Z80* (Sybex), contient les synchronisations pour toutes les instructions Z80. Voici les fréquences d'horloge de l'UC de machines basées sur Z80 : ZX81 (3,25 MHz); Spectrum (3,5 MHz); Tandy TRS80 (1,7 MHz).

Pour obtenir un très petit retard, on peut utiliser l'instruction NOP. Cette instruction, sur un micro à 2 MHz, donnera un retard de deux microsecondes. On peut en utiliser un certain nombre successivement, mais des retards plus longs peuvent être obtenus en appelant des routines factices; par exemple, la routine suivante donnera un retard de 27 T :

```

CALL DELAY
RET
  
```

Dans cet exemple, l'instruction CALL prend 17 T, et RET en prend 10. Ainsi, avec un processeur à 2 MHz, le retard sera de  $13,5 \mu s$ . Pour augmenter légèrement ce retard on peut inclure des instructions NOP au début de la routine.

Pour des retards plus importants, il faut utiliser une boucle. Dans l'exemple suivant, un registre est chargé avec une valeur, laquelle est ensuite décrémentée dans une boucle. La routine donne un retard de 99 T (soit  $49,5 \mu s$  à 2 MHz).

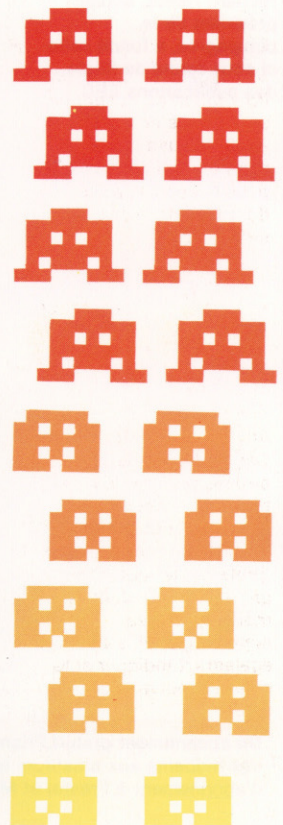
INSTRUCTION	TEMPS PRIS (ÉTATS T)
CALL DELAY	17
(DELAY LOOP)	
LD B,5	7
DEC B	4
JR NZ, LOOP	12 (ou 7 si vrai)

Les trois instructions commençant par LD B,5 sont la boucle de retard (delay loop) proprement dite. Comme dans une routine en langage machine 6502, la durée totale de cette routine varie suivant la valeur chargée dans le registre. Le nombre total de cycles d'horloge qu'elle prend pour effectuer ce code peut s'exprimer par :  $C = 24 + (N \times 16) - 5$ , où N est la valeur chargée dans le registre B. On peut aussi utiliser des compteurs de boucles emboîtées. Mais ici nous devons tenir compte d'autres considérations. Primo, tout registre utilisé pendant une telle routine doit être mis sur pile pour préserver son contenu. Secundo, certaines machines ont des interruptions matérielles qui modifieront la synchronisation. Les interruptions masquables sont désactivées et réactivées par les instructions DI et EI. La routine suivante utilise des compteurs de boucles emboîtées :

INSTRUCTION	FONCTION	TEMPS (T)
DI	Désactive interruptions	4
PUSH DE	Préserve le contenu du registre	11
LD D,n	Valeur du compteur interne	7
LD E,n	Valeur du compteur externe	7
CALL OLOOP	Saut au compteur externe	17
POP DE	Restaure contenu	10
EI	Réactive interruptions	4
:		
(OLOOP)		
DEC E	Décrémente boucle externe	4
RET Z	Fin si zéro	11 ou 5
(ILOOP)		
DEC D	Décrémente boucle interne	4
JP Z,OLOOP	Saut si D = 0	10
JP ILOOP	Sinon continuer boucle interne	10

Dans cette routine, le retard est augmenté si la valeur du registre E l'est. La routine se terminera lorsqu'on décrémente le registre E, et que le résultat s'annule. Si la boucle interne atteint zéro et que la boucle externe garde une valeur supérieure à un, la boucle interne sera initialisée à 255, et décrémentée jusqu'à zéro avant que le contrôle revienne à la boucle externe.

**Invasion synchronisée**  
Les retards en langage machine sont indispensables dans les programmes de jeux, en particulier lorsqu'il y a un objet mobile sur l'écran avec lequel le joueur doit interagir. Un exemple classique en est le jeu Space Invaders. Sans retard, le mouvement des envahisseurs ennemis serait trop rapide. Avec des retards soigneusement programmés, les mouvements peuvent être suffisamment contrôlés pour rendre le jeu intéressant.





# De toutes les couleurs

**La société Prism est un exemple en Grande-Bretagne. Elle a joué un grand rôle dans la mise sur pied du réseau Micronet. Elle s'oriente actuellement vers la diffusion des robots domestiques.**



**Maître à bord**  
Richard Heath, actuel président de la compagnie, a fondé celle-ci sous forme de filiale des publications EEC.

La plupart des compagnies liées à l'informatique travaillent à très court terme, et se bornent à satisfaire les besoins courants en matériel et en logiciel. Mais Prism, en Grande-Bretagne, voit plus loin. Outre un catalogue très important du matériel, la firme a été l'un des créateurs de Micronet — la première grosse base de données accessible aux simples particuliers — et a déjà commencé la commercialisation de robots bon marché.

La compagnie a été créée en 1982 par les publications EEC afin de mettre sur pied Micronet. Richard Heath et Bob Denton en devinrent les directeurs. Micronet passe par Prestel (équivalent à notre Télétel en France), le plus gros système vidéotex de Grande-Bretagne, afin de proposer aux usagers toutes sortes de services : transfert de programmes, accès aux informations, échange de « courrier électronique ». Les publications EEC avaient déjà lancé *Sinclair User*, une revue mensuelle, alors que les produits de Sir Clive n'étaient encore disponibles que par correspondance, ou chez les détaillants de la chaîne WH Smith. Le magazine connut un succès prodigieux, en dépit du scepticisme de Terry Cartwright, aujourd'hui directeur du marketing de Prism : « Je pensais alors que ce ne serait qu'un feu de paille. Mais quand nous sommes allés à la première ZX Microfair, nous avions emmené 8 000 bulletins d'abonnement et les gens faisaient la queue tout autour de l'immeuble : en quelques heures nous avons épuisé tous nos bulletins. »

Sinclair ayant ensuite décidé d'approvisionner les détaillants, Prism signa un contrat avec lui afin d'assurer la distribution du ZX81 et du plus récent Spectrum. En fait, le nom de la compagnie avait été délibérément choisi pour

évoquer dans l'esprit du public l'idée d'une association étroite : un rayon de lumière qui passe à travers un prisme (*Prism*) fait naître tout un spectre (*Spectrum*) de couleurs ! La firme déclarait récemment avoir vendu plus de 500 000 appareils, soit près du quart du marché anglais de la micro-informatique.

En mars 1983, Prism lança Micronet. La société s'occupa essentiellement du matériel, proposant des modems (fabriqués par OE Ltd. et Thorn EMI) pour les ordinateurs les plus répandus. Elle en a récemment commercialisé un destiné au Commodore 64.

Micronet a déjà près de 10 000 abonnés ; mais Prism a récemment vendu les parts qu'elle détenait dans le réseau, et se concentre actuellement sur la distribution et la vente du matériel. Outre le Spectrum, elle met en vente l'Oric et l'Atmos, ainsi que le Wren, un ordinateur de gestion portable créé par son propre service de recherche. Fabriqué par Thorn EMI, l'appareil est enfin disponible, après des retards liés à la production.

Prism se tourne aussi vers un secteur en rapide expansion, celui des robots domestiques. Elle vend aussi bien « Topo », un modèle d'origine américaine (plus de 20 000 F) que des kits très bon marché, les « Movits » (entre 200 et 500 F environ).

Terry Cartwright explique : « Les robots soulèvent un intérêt énorme. Je ne sais pas ce que les gens en feront, mais après tout, en 1976, personne n'avait la moindre idée de ce qu'on pourrait bien faire avec un Apple. » Prism espère par ailleurs distribuer le QL, le tout nouvel ordinateur Sinclair. Cette politique de diversification — et d'expansion à l'étranger — devrait se poursuivre dans les années à venir.

**Un abonnement gratuit.** Prism compte louer ces modems aux amateurs leur offrant un an d'abonnement à Prestel et Micronet.



### Premiers pas

Le Wren est un ordinateur de gestion construit par Prism autour du Z80. Il est pourvu de deux lecteurs de disquettes et d'un modem intégré qui permet de le connecter à Prestel.



# abc

## INFORMATIQUE

Pour classer et présenter dans votre bibliothèque les fascicules de votre collection, des reliures mobiles pratiques et élégantes sont en vente chez tous les marchands de journaux.

Dans chaque reliure, vous trouverez, dans l'enveloppe qui contient les deux lames métalliques de la reliure, un décalque portant les numéros 1 à 8, qui vous permettra de marquer vous-même le dos de chaque volume. Pour relier les 12 fascicules qui composent un volume, vous devrez en retirer les couvertures sans endommager les agrafes métalliques. Le numérotage de la reliure puis la mise en place des fascicules doivent être effectués selon les instructions ci-dessous.

1

Disposez à plat la reliure. Enlevez le papier de protection du décalque. Positionnez le décalque en faisant coïncider l'écran qui entoure le numéro choisi avec l'écran situé sur le dos de la reliure.

2

Avec la pointe d'un stylo à bille, frottez régulièrement le numéro à transférer, en exerçant une certaine pression et en débordant légèrement.

3

Enlevez doucement le support : le numéro est reporté sur la reliure. Posez dessus le papier de protection du décalque et frottez largement avec un objet poli ou arrondi, de manière à assurer la parfaite adhérence des caractères transférés.

4

Retournez la reliure mobile. Introduisez, d'un côté seulement, dans les encoches pratiquées dans l'épaisseur de la couverture, l'une des extrémités des deux lames d'acier livrées avec la reliure.

5

Vérifiez le bon ordre des fascicules, puis passez les deux lames d'acier dans les agrafes supérieures et inférieures.

6

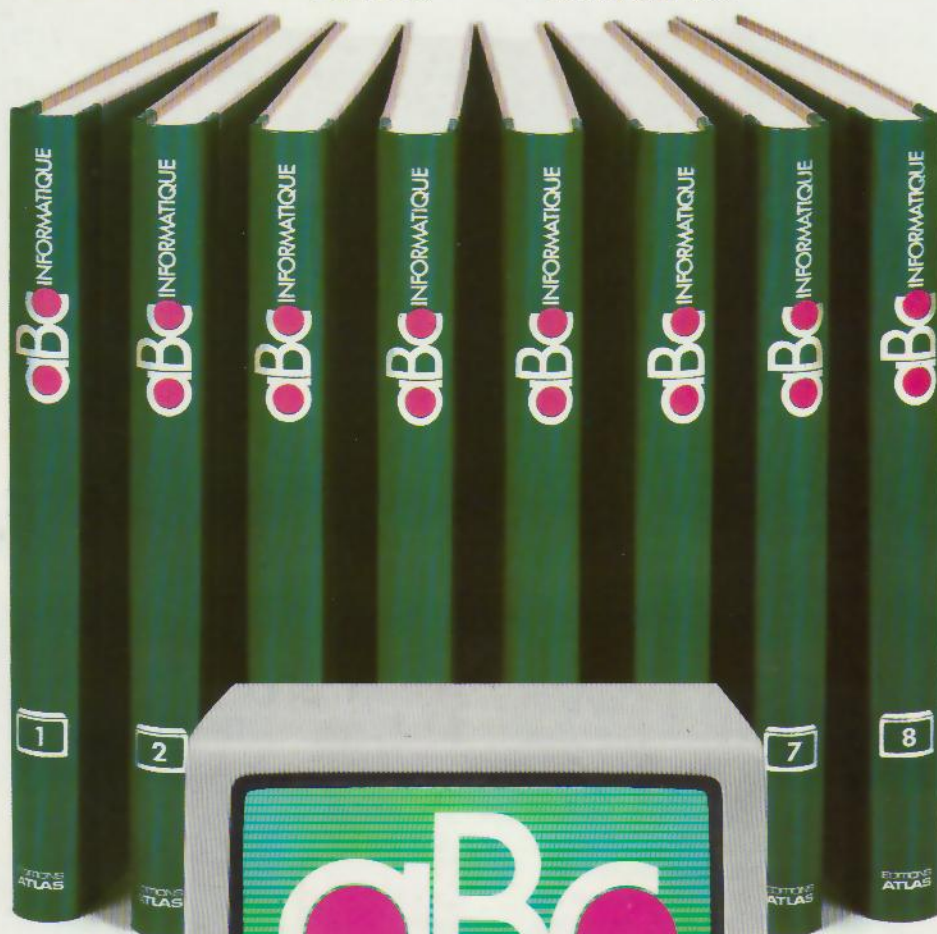
Introduisez les extrémités libres des lames d'acier dans les encoches correspondantes de la reliure.

Ce système original, sans mécanisme visible, donne au volume l'apparence d'une reliure classique. Nous vous conseillons d'avoir toujours une reliure d'avance pour mieux protéger vos fascicules au fur et à mesure de leur parution.

# Protégés par une élégante reliure, vos numéros d'aBc Informatique seront plus faciles à consulter

Pour classer, répertorier, protéger vos fascicules d'ABC Informatique, les Editions Atlas vous proposent des reliures élégantes, sobres, qui s'insèrent parfaitement dans votre bibliothèque. Chacune contient 12 fascicules,

les maintient, les préserve. Un système simple, résistant, vous permet de les assembler facilement. Elles sont en vente en permanence chez votre marchand de journaux. Demandez-les!



Chaque  
reliure:

**40 FF**  
**295 FB**  
**18 FS**

Cod. N.M.P.P.:  
6103

EDITIONS  
**ATLAS**